Mathematical Tables
and other
Aids to Computation

A Quarterly Journal

Edited by

VII
NOS. 41–44
1953

Published by

THE NATIONAL RESEARCH COUNCIL
Washington, D. C.

M

Ai

Mathematical Tables

and other

Aids to Computation

A Quarterly Journal

VII • Number 41 • January, 1953 • p. 1–72

Published by

THE NATIONAL RESEARCH COUNCIL
Washington, D.C.

SUBSCRIPTION RATES

1943–1945: (Nos. 1–12) $12.00 for all 12 issues (not available for separate sale except as noted below*)

1946–1949: $4.00 per year

1950–1953: $5.00 per year

Single issues are available for sale as follows:

* 1944 (No. 7, "A Guide to Tables of Bessel Functions," by H. Bateman and R. C. Archibald, 104 pp.) $2.00

1946–1949 (Nos. 13–28) $1.25 for single issue

1950–1953 (Nos. 29–40) $1.50 for single issue

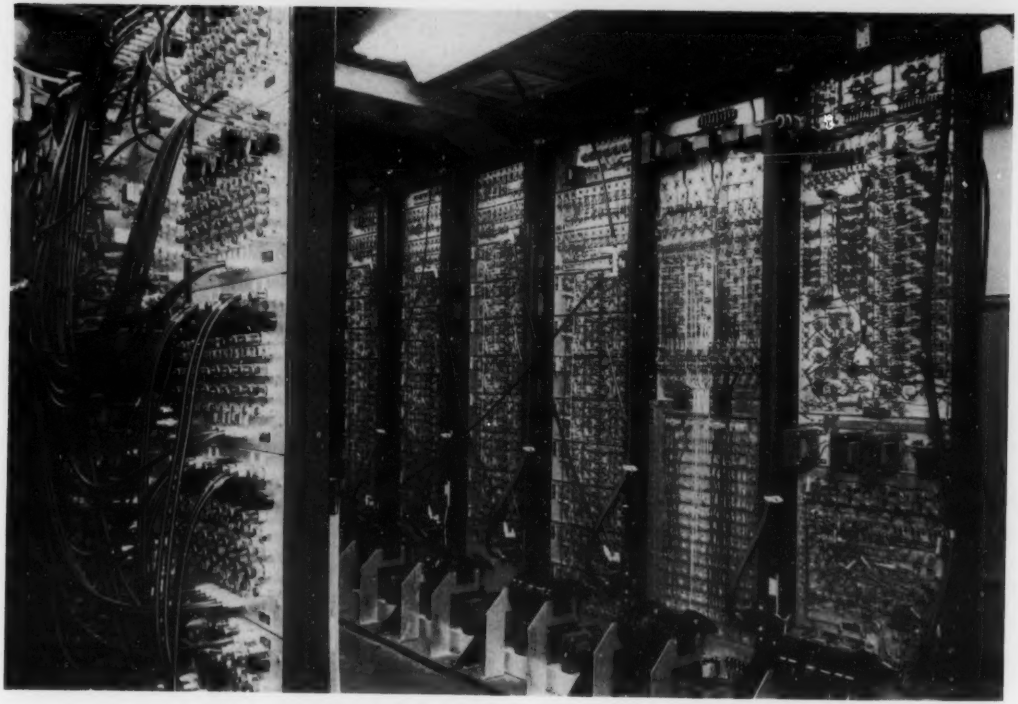
All payments are to be made to National Academy of Sciences, 2101 Constitution Avenue, Washington, D. C.

Agents for Great Britain and Ireland (subscription 42s, 6d for 1953) Scientific Computing Service, Ltd., 23 Bedford Square, London W.C.1.

Published quarterly in January, April, July and October by the National Research Council, Prince and Lemon Sts., Lancaster, Pa., and Washington, D. C.

All contributions intended for publication in *Mathematical Tables and Other Aids to Computation*, and all Books for review, should be addressed to D. H. Lehmer, 16556 Chattanooga Place, Pacific Palisades, Calif.

Entered as second-class matter July 29, 1943, at the post office at Lancaster, Pennsylvania, under the Act of August 24, 1912.



THE USAF-FAIRCHILD SPECIALIZED DIGITAL COMPUTER

Addendum to a Guide to Tables on Punched Cards

A Guide to Tables on Punched Cards was published in October, 1951 [*MTAC*, v. 5, p. 185–212]. In July, 1952 [*MTAC*, v. 6, p. 204–205], an announcement was made about a forthcoming addendum to this guide, and information was requested concerning any new contributions or necessary corrections. This addendum and list of corrections is now given below. In addition, information is supplied in Section 2 about keypunching of important tables that is now in progress at various laboratories.

It might be interesting to observe that the supply of new tables on punched cards is rather small, compared with the activity in mathematical computing that has been stimulated by the increasing number of high-speed calculators. It is possible that this foreshadows the replacement of punched cards by more satisfactory tools, which may become available even to laboratories of relatively small size. For the present, however, punched cards are still very much in use and it is deemed desirable to keep the record of them up-to-date.

Additions:

Source	Available at		Description of Tables
[(18)]	(18)C	2.6	$\sqrt{x}$, $\sqrt{x/10}$: $x = 0(.01)6(.1)14.6$; 5D
[(18)]	(18)C	2.6	$\sqrt{x}$, $\sqrt{x/10}$: $x = 0(.001)2$; 5D
[(18)]	(18)C	2.6	$\sqrt{x}$: $x = 1(1)9999$; 9D; 4000 cards (x = 0001, 1001, 2001 on same card, etc.)
[(24)]	(24)C	2.9	Sums related to powers of complex numbers. $U_n = z^n/n!$; $T_n = U_n/n$ $$R_k + iS_k = \sum_{n=1}^{\infty} U_{k+4n},$$ $$P_k + iQ_k = \sum_{n=1}^{\infty} T_{k+4n},$$ $R_k, S_k, P_k, Q_k, k = 1(1)4$; 10D $z = x + iy; x, y = 0(.1)3.1$; $y \leq x$. Useful for computing the exponential functions and their integrals in the complex plane.

Source	Available at		Description of Tables
[(24)]	(24)C, (73)UMT 145[D,F]	7.1	Cyclotomic cosines $2\cos(2\pi k/p)$: $k = 1(1)\left(\frac{p-1}{2}\right)$; 20D p an odd prime < 100. 517 values (Computed under supervision of D. H. LEHMER.)
[(67)]	(67)	7.4	$\sin x$, $\cos x$, Δ, $\Delta/60$: $x = 0(1')90°$; 8D (See Note 1 at end of section.)
[(67)]	(67)	7.4	$\sin x$, $\cos x$, D^1, D^2: $x = 0''(1000'')129600''$; 8D (See Note 1 at end of section.)
[(67)]	(67)	7.5	$\sin x$, $\cos x$, Δ: $x = 0^s(100^s)172800^s$; 5D
[(67)]	(67)	7.5	$\sin x$, $\cos x$, Δ: $x = 0(0^h.01)11^h.99$; 4D
[(67)]	(67)	9.10	$y = \arcsin x$, D^1, D^2: $x = 0(.01)0.52$; y to 6D of a degree
[(67)]	(67)	9.15	$y = \arctan x$: $x = 0(.01)1$; y to 7D of an hour
[(23)–(18)]	(18)C	10.0	e^{-x}: $x = 0(.01)30$; 14D
[(18)]	(18)C	10.0	e^{-x}: $x = 0(.001)2$; also $x = 0(.01)6(.1)14.6$; 5D
[(16)–(4)]	(4)	14.61	$\frac{1}{\Gamma(z)} = U + iV$: $z = x + iy$; $x = -.5(.01).5$; $y = 0(.01)1$; 6D
[(25)]	(25)	14.9	Incomplete Beta function $\beta_x(p, q)$: $x = 1$, $p \leq q = 0.05(.05)10$; 6S $x = \frac{1}{2}$; $p = q = .05(.05)10$; 6S See Bureau of Mines R I 4917, in press, for details and tabular entries.
[(5)]	(18)C	17.1	$J_0(x)$, $J_1(x)$, Δ, Δ^2, Δ^3, Δ^4: $x = 25.01(.01)99.9$; 18D
[(66)–(4)]	(4)	18.1	$I_0(x)$, δ^2: $x = 0(.001)5$; 9S
[(66)–(4)]	(4)	18.3	$K_0(x)$, δ^2: $x = .01(.01)5$; 8S
[(66)–(4)]	(4)	{18.3, 18.4}	$e^{-x} I_0(x)$, δ^2, $e^x K_0(x)$, δ^2: $x = 5(.01)10(.1)20$; 8S

Source	Available at		Description of Tables
[(24)]	(24)C	**20.294**	$f(x) = \int_0^x Ai(-t)dt$ $F(x) = \int_0^x f(t)dt$ where Ai(t) is the familiar Airy integral: $x = -2(.01)5$; 8D in $f(x)$; 7D and 8D in $F(x)$, with the eighth place uncertain. Computed by ELMER E. OSBORNE.
[(24)–(12)]	(24), (12)C	**22.56**	$e^{-x} L_n(2x)$, where $L_n(2x)$ is the Laguerre polynomial: $n = 0(1)12$, $x = 0(.1).5(.5)10(1)20$; 10D $n = 13(1)20$; $x = 0(.5)10(1)20$; 10D Computed under supervision of C. LANCZOS.
[(4)]	(4)C	**22.68**	Fermi functions (see definitions in note 2 at end of section). *Table 1.* $\phi_0(p, z)$: $\|z\| = 2(1)96$; $p = 0(.1)1(.2)3(.5)5(1)9(2)15$ $\|z\| = 2(1)25$; $p = 20, 25$; to about 4S *Table 2.* $\frac{p}{W}\frac{F_\nu}{F_0}$: $\nu = 1, 2, 3$; same range as Table 1, except that $p \geq 0.1$.
[(15)]	(15)	**25.1**	Random digits. Random with respect to card column, as well as with respect to digits in col. Made by JACK SHERMAN. See Note 4.
[(18)]	(18)C	**25.1**	8-digit random numbers computed by Lehmer's rule—9 numbers per card, 3,000 cards
[(4)–(4)]	(4)	**25.21**	Random variates from distribution $\exp(-y)$. [A number r was chosen at random from 4-decimal numbers. Then $-\log r = y$]. 140,000 cards.
[(14)–(4)]	(4)	**25.31**	Random direction cosines from a spherically symmetric distribution. Three-dimensional, l, m, n, to 3 decimals. 70,000.

Source	Available at		Description of Tables
[(25)]	[25]	**26.1**	Tables relating to hydrodynamics Tables of velocity of steady laminar flow in channels of rectangular cross-section (see definition in note 3 at end of section). $w(x, y, \sigma)$ to 6D for x, y, σ ranging between 0 and 1.
Northrop	(73)UMT 138[F]	**27.1**	Products of powers of small primes $N = 2^a\, 3^b\, 5^c\, 7^d$: $a = 0(1)11$; $b = 0(1)8$; $c = 0(1)5$; $d = 0(1)4$; exact. Computed by R. A. JOHNSON, Northrop Aircraft, Inc., Hawthorne, Calif.
[(24)]	(24)C, UMT 146[F,L]	**27.7**	Kloosterman sums $s_p(k) = \sum_{n=1}^{p-1} \exp\{2\pi i(kn + \bar{n})/p\}$ $n\,\bar{n} \equiv 1 (\mathrm{mod}\ p)$ $k = 1(1)\ (p-1)$, p a prime < 100, 19D. 1034 cards. Computed under supervision of D. H. LEHMER.
[(18)]	(18)C	**28.1**	Astronomy. Julian date to calendar date 1639 (30 Julian days) 2060; 3,846 cards

Note 1. The Admiralty Computing Service is now at an end; but the punched cards listed under [67] are available.

Definitions: $D^2 = \frac{1}{4}(\delta_0^2 + \delta_1^2)$; $D^1 = \delta_{\frac{1}{2}} - D^2$.

(If h is the interval between two consecutive arguments, then $f(x_0 + ph) = f_p = f_0 + pD^1 + p^2D^2$.)

Note 2. Fermi functions 22.68

$$F_\nu = \left[\frac{(2\nu+2)!}{\nu!}\right]^2 (2p\, r_0)^{2(s_\nu - \nu - 1)} \frac{e^{\pi y}|\Gamma(s_\nu + iy)|^2}{\Gamma^2(2s_\nu + 1)};$$

$$\phi_0\,(p, z) = \frac{p}{W}\left(\frac{1+s_0}{2}\right) F_0\,(p, z);$$

$s_\nu = \sqrt{(\nu+1)^2 - \alpha^2 z^2}$; $y = \alpha\, z\, W/p$; $W = \sqrt{p^2+1}$; $r_0 = \frac{1}{2}\alpha\, A^{\frac{1}{3}}$; $\alpha = 1/137.03$

z = atomic number of β emitter; $z > 0$ for electrons; $z < 0$ for positrons.

A = mass number of β emitter; p = momentum of β particle in mc units
W = total energy in mc^2 units; r_0 = nuclear radius in h/mc units.
Write to (4) for further details.

Note 3. Tables relating to hydrodynamics 26.2

$$w(x, y, \sigma) = 1 - x^2 + \frac{4}{\pi^3}\sum_{n=0}^{\infty}(-1)^{n+1}(n+\tfrac{1}{2})^{-3}\cos(u_4)\,F(n, y, \sigma)$$

where

$$F(n, y, \sigma) = (\exp u_1 + \exp u_2)/(1 + \exp u_3)$$

$$u_1 = -(n+\tfrac{1}{2})(1-y)\frac{\pi}{\sigma};\ u_2 = -(n+\tfrac{1}{2})(1+y)\frac{\pi}{\sigma};\ u_3 = -(2n+1)\frac{\pi}{\sigma};$$

$u_4 = (n + \frac{1}{2})\pi x$. See Bureau of Mines R.I. 4885 (July 1952) for further details.

Note 4. The random digits were prepared by W. F. Brown, Jr., of Sun Physical Laboratory, Newton Square, Pa., with the aid of random digits of the Rand Corp.

Corrections: [*MTAC*, v. 5, 1951]

p. 190, Source 55(b), *for* sinh $F - 1$ *read* sinh $F - F$.

p. 196, **7.1** Items 2 and 10 under this listing, (18)C: *for* $x = 0(.001)1.571$; 7D *read* $x = 1(.001)1.228$.

p. 201, **10.0** Item 9, l. 13. *for* 14S and 18S *read* 15S.

p. 201, **11.6** Add the remark: "See also **28.3**."

p. 203, **14.6** *for* $x = 9(.1)10$; $y = 0(.1)10$; 14D or 15D *read* $x, y = 0(.1)10$; 12D and 14D. (Extension previously in progress now completed.)

p. 205, **17.1** Item 3, add "Available at (18)."

p. 209, **25.2** Tabulation of x^2 not available at (4).

p. 209, **25.3** *for* Cards 0(1)14,566 *read* Cards 0(1)13,000.

p. 209, **25.3** *for* 9.4247 *read* 1.5707.

p. 211, **28.2** Kepler's equation: *for* $E = M - e \sin E$

read $M = E - e \sin E$.

Keypunching in progress.

(a) At [24]: $I_\nu(x)$, $\pm\,\nu = \frac{1}{3}, \frac{2}{3}, \frac{1}{4}, \frac{3}{4}$

(b) At Point Mugu, under supervision of Donald Dufford, NAMTC, $J_\nu(x)$, for same values of ν as those above. Thus both volumes of the NBS Tables of Bessel functions of fractional order are now being keypunched.

(c) At [24]: $I_n(x)$, based on manuscript of BAAS now in print.

(d) [24]: Spherical Bessel functions. Keypunched but not checked.

(e) At NOTS, KENNETH C. RICH and CHARLES RICKER, China Lake Pilot Plant, China Lake, California. Bessel functions $y_n(x)$ and $Y_n(x)$, $n \leq 20$, BAAS, v. 10.

Readers are again requested to review their keypunching loads, for possible keypunching of other basic mathematical tables during spare hours. Any one who can undertake a part of this work is requested to communicate with the undersigned.

GERTRUDE BLANCH
EVERETT C. YOWELL

National Bureau of Standards
Los Angeles 24, Calif.

The Sieve Problem for All-Purpose Computers

Introduction. The term all-purpose digital computer is often used to indicate a computing machine capable of performing the rational operations using addition and multiplication as basic functions of the arithmetic unit. Even when these operations are supplemented by some discrimination and "extract" commands, for dealing directly with the digits of numbers, there are a number of processes to which such a machine is not well suited. This includes even finite processes that are combinatorial in nature. Perhaps the most well-known of these processes is the sieve process. Although special equipment has been constructed for carrying out this algorithm[1] we shall not describe it here. Our purpose is to indicate how the all-purpose computer may be programmed to compete with sieve machines.

Let us first state the general problem to be solved by the sieve process. Let $m_1, m_2, \cdots, m_k$ be a set of k positive integers which, for the purposes we have in mind, may be assumed to be relatively prime in pairs. For each m_i we consider n_i distinct arithmetical progressions or linear forms in the variable x which we denote by

$$P_{ij}(x) = m_i x + a_{ij} \begin{cases} i = 1, 2, \cdots, k \\ j = 1, 2, \cdots, n_i \end{cases}.$$

We may assume that, for i fixed, the a_{ij} are distinct non-negative integers less than m_i. The problem is to find all integers N between given limits, say

$$A \leq N < B,$$

such that each N belongs to k arithmetical progressions.

The number k is called the *width* of the problem, the k numbers m_i will be called the *moduli* of the problem and the numbers a_{ij} ($j = 1 \cdots n_i$) the *admissible remainders* for the modulus m_i. The solution N, or even the number of solutions, is an exceedingly complicated function of the given parameters m_i, a_{ij}, A, B. There are two extreme cases which may be mentioned.

Eratosthenes Sieve Problem. In this case $m_1 = 2$, $m_2 = 3$, and in general m_i is the i-th prime number, k is the number of primes not exceeding $A = B^{\frac{1}{2}}$, $n_i = m_i - 1$ and $a_{ij} = j$. This gives the famous sieve of Eratosthenes and has for solutions (besides $N = 1$) all the prime numbers between $B^{\frac{1}{2}}$ and B. In this sieve there is the maximum number of admissible remainders for each modulus.

Chinese Sieve Problem. In this extreme case there is the minimum number of admissible remainders for each modulus, that is, $n_i = 1$ for all i. This problem is very old[2] and often "solved" by what is known as the Chinese Remainder Theorem. If $A = 1$ and B is the product of the moduli, then there is a single solution N which may be found by any one of a number of other nearly equivalent practical methods rather than by a bona fide sieve process. On the other hand, this type of sieve problem is often useful as a checking routine for a more general sieve setup.

Quadratic Sieve. Midway between the two extreme examples we have examples in which each n_i is approximately $\frac{1}{2}m_i$ so that, to put it roughly, any number N has, a priori, an even chance of belonging or not belonging to one of the arithmetical progressions of a fixed modulus m. The expected number of solutions N is therefore approximately $(B - A)/2^k$. This kind of problem is the one most frequently met with and occurs in problems involving quadratic residues and congruences, binary quadratic forms, Diophantine equations of the second degree, etc.; hence the reason for calling this the quadratic sieve problem.

Theoretical Aspects. It is theoretically possible to combine any two arithmetical progressions with relatively prime moduli

$$P_1 = m_1x + a_1 \qquad P_2 = m_2x + a_2$$

into a single one of the form

$$P_3 = m_1m_2x + a_3$$

where P_3 consists of all numbers common to P_1 and P_2. Thus all numbers belonging to both the forms

$$5x + 4 \qquad 12x + 5$$

comprise the arithmetical progression

$$60x + 29.$$

By proceeding in this way one may combine the k sets of arithmetical progressions into a single set whose modulus is $m = m_1m_2 \cdots m_k$. The number of arithmetical progressions in this one set is clearly $n = n_1n_2 \cdots n_k$. The ratio n/m is the density of solutions of the problem. The above process becomes intolerably unwieldy even for moderate values of k except when most of the n_i are equal to unity. In the quadratic case, for example, a moderate value of k, like 20, which eliminates all numbers but one in a million, would lead to one set of at least $4 \cdot 10^{21}$ arithmetical progressions with a modulus of $4 \cdot 10^{27}$. To produce and sort for size the $4 \cdot 10^{21}$ admissible remainders is clearly out of the question. In these cases it may be, indirectly, more practical to examine all integers N between A and B excluding each integer in turn for non-membership in one of the original sets of arithmetical

progressions. This is the procedure we speak of when we refer to a sieve process.

Normalized Sieve Problem. In order to make a comparison of the effectiveness of different sifting processes we restrict ourselves to the case in which $n_i > 1$ for all i. That this is no real restriction is seen from the following consideration. If, for example, $n_1 = 1$ so that we are looking for an N of the form $m_1x + a_{11}$, we may change variable from N to N_1 where $N = m_1N_1 + a_{11}$ and thus eliminate $P_{11}(x)$. This reduces the width of the problem by unity, introduces new constants a_{ij} in the remaining $k - 1$ sets of progressions, but leaves the other n_i unaltered. The limits A and B are replaced by A/m_1 and B/m_1 so that the result is a fictitious speed-up of the sieve process by a factor of m_1. Proceeding in this way with any other cases of $n_i = 1$ we finally come to grips with the real problem in which $n_i > 1$. To be sure, a case in which $n_i = 2$ breaks up the problem into two parts each of which may be considered separately as a problem with an $n = 1$. The result is two short problems instead of one long one. This dissection of the problem is a favorite method with hand computers and accounts for the fact that fairly slow automatic sieves have considerable competition from hand methods. Of course the same dissection method is applicable to automatic sieves too, but often it is not practical because of the comparatively long set-up time involved. For these reasons, therefore, we consider as a normalized sieve problem, one in which $n_i \geq 2$. To put the matter in another way, we consider that all integers N between A and B have, a priori, an equal chance of satisfying the conditions imposed by the problem. The effectiveness of a particular process will then depend only upon the speed with which it can dispose of the average candidate N and pass on to $N + 1$.

Rates of Special Sieves. Without going into details it is desirable for purposes of comparison to mention existing sieves and to give an idea of their effectiveness. First of all we have hand methods. These involve the so-called movable strip method and its generalization, the stencil method. Here the range of natural numbers is represented by one or more sheets of paper ruled in squares, each square representing, by virtue of its position on the sheet, a definite integer. Strips of paper of length m_i are punched with $m_i - n_i$ holes, representing the inadmissible remainders modulo m_i, and are moved carefully down the columns of the sheets and those square cells which are revealed by the punched holes are crossed out by the computer. After k of these strips have been applied, those cells which still survive are the answers to the sieve problem. Rates as high as 3 numbers per second are difficult to maintain over a long period of time.

Electromechanical sieves have been built to canvass 50 numbers per second. These sieves can accommodate almost any reasonable sized moduli m_i and handle problems of width $k \leq 20$.

A photo-electric sieve has been built in 1932 which runs at 6000 numbers per second and an electronic sieve is under development which is designed for approximately 300000 numbers per second. These high-speed sieves are not equipped with high-speed output and are intended to be used on problems whose progressions give restrictions comparable to or higher than those of quadratic type. It will be seen that such high speeds are not obtainable

with all-purpose computers but that the more modest speeds of the electromechanical type sieve can be achieved and even surpassed.

Direct Attack by an All-Purpose Computer. Most all-purpose computers are equipped with a division command or at least a division program. A direct method of handling the sieve process would consist in dividing a trial value of N by each modulus m_i in turn and then inspecting the remainder for membership in the set α_{ij}. In the case of the Eratosthenes sieve, the inspection of the remainder is practically instantaneous as it requires only one or two addition times. If the number of moduli is large, for example large enough to make a list of primes above 10^7, then the number of divisions in case N is a prime will be greater than 440. Speeds are such that 50 milliseconds per division is not often realized. Allowing only 10 milliseconds per division, times of the order of several seconds are required to treat these cases. This does not compare favorably with even hand methods. However, in case the number of moduli is small a much more favorable speed prevails. A small Eratosthenes sieve is often incorporated into a problem in which a parameter p is supposed to have a prime value. In this way a large percentage of composite p's are eliminated, the remaining composites being taken care of later either from the output or by some other internal program. Beyond a certain point we get diminishing returns from each new modulus we introduce.

In the case of a typical quadratic sieve problem the examination of the remainder would be a much longer routine but still, on the average, rather shorter than a division program. Sieves of width 25 would canvass numbers at the rate of three or four per second. This is still too slow to compete with much simpler equipment. Besides, the memory capacity of many machines would be exceeded in trying to accommodate the given remainders a_{ij} in many typical problems.

Binary Method of Representing a Sieve Problem. The fact that, as far as a given modulus m is concerned, a proposed number N is either rejected or accepted, makes it possible to treat the sieve with binary methods. The sieve problem may in fact be represented by a matrix of k rows and infinitely many columns. The infinite rows are periodic, the i-th row having a period of m_i. The element situated in any column whose number is congruent to r modulo m_i is 0 or 1 according as r is or is not one of the acceptable remainders a_{ij}. The problem is then to find those columns which consist wholly of zeros. (Clearly the rôles of 0 and 1 may be interchanged here.) The matrix will be referred to as the sieve matrix.

For example, suppose that the problem is to find the least positive integer of the forms

$$\begin{cases} 5x + 1 \text{ or } 4 \\ 7x + 2 \text{ or } 3 \text{ or } 4 \\ 9x + 1 \text{ or } 6 \text{ or } 7 \end{cases}$$

The matrix corresponding to this problem is

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & \cdots \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & \cdots \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & \cdots \end{bmatrix}$$

Here we are looking for the first column which consists wholly of zeros.

The answer to the problem will be seen to be 16. In fact if we continue each periodic row two more steps beyond what is shown, we obtain a column of zeros.

Machine Methods of Realizing the Sieve Process. Most of the high-speed digital computers use binary numbers and almost all of the decimal type machines use a binary code for decimal digits. The purely binary machines are the most natural ones with which to apply the following methods. The use of decimal type machines entails a considerable reduction in the width of the problem or else a certain amount of indirect use of the binary code for the decimal digits. In what follows we shall assume that we are dealing with a binary type machine, leaving the modifications necessary for the decimal machine to the ingenuity of the reader. The operations needed to carry out the sieve process clearly involve shifting operations and the extraction or discrimination of digits. This does not mean that addition, subtraction, and multiplication are not needed. In fact these latter operations are frequently involved in the execution of the former ones.

Of the various possible methods to be used that one is to be preferred which gives the greatest speed in canvassing the columns of our matrix. The decision may be determined by the logical structure of the machine. In any case, there are two general steps to carry out: (a) the passage from one column of the matrix to the next and (b) the inspection of a column to see if it contains a non-zero element.

First Method. The steps (a) and (b) can be carried out simultaneously by the simple process of doubling the binary number represented by each row. Since in many cases the period m_i will exceed the number s of digits which the adder will accommodate, it will be necessary to program this doubling as a "multiple precision" process, taking account of overflow between the separate batches of s digits of the entire number. In particular, if overflow occurs in the first column there are two important consequences. First, this indicates that a 1 was standing in the first column and so the corresponding number N is to be rejected. Secondly, this overflow must be sent to the other end of the period to perpetuate the periodic pattern of the digits. This operation has to be performed k times, once for each row. If at any time no overflow is obtained in the entire set of k operations, then we have a solution of the problem and the machine is instructed to print out the answer, continuing on or halting as desired. The entire process can be carried out using only addition with detection of overflow, and involves a certain amount of "tallying" and modification of commands. The method has the disadvantage of requiring k multiple precision operations in passing from N to $N + 1$. When k is large (20 or 30 in some typical cases) and the m's large, the major portion of the time is spent in keeping the periods up-to-date. However, for small k and m, the method is both simple and fast. In fact, its use is recommended in problems in which no actual sifting is involved. Suppose, for example, that we have a problem involving one or more parameters and these are given certain irregularly spaced values such as for instance

$$x = 1, 3, 6\text{–}9, 12, 15, 18\text{–}22, 25.$$

Instead of storing all these numbers in as many memory positions, this

information may be stored in only the one word

1 0 1 0 0 1 1 1 1 0 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1,

in which the r-th digit from the left is 1 or 0 according as r belongs to our set or not. By successive doubling and inspection for overflow the machine can inform itself as to which values of x it should consider.

Second Method. In order to eliminate the lengthy multiple precision processes of the first method one may proceed as follows. The periodic digits of each row of our matrix are stored as before in "words" of s digits. The minor of k rows and the first s columns of the original matrix are now duplicated in a special part of the memory where the operation of doubling and testing for overflow is performed as in the previous method. However, when overflow occurs the fact is recognized by the machine but the digit is not kept track of as before. After s doublings the information in the original k by s matrix is all used up and the matrix now consists wholly of zeros.

The next process consists of circulating the original matrix by moving the elements of each row s places to the left. Since s binary digits is a word, this process is simple and fast. Only one complication arises, that of disposing of the first word in each row. If for a particular modulus m_i we have

$$m_i = sq_i + r_i \qquad (0 \leq r_i < s),$$

then the first $s - r_i$ digits of this first word w_i belong in the penultimate word of the shifted matrix while the last r_i digits form the last word. This disposal of w_i can be accomplished by use of a product command. The machine computes the exact two word product of w_i by 2^{s-r}. The "most significant" word of this product is thus added to the penultimate word of the shifted matrix and the least significant word is placed at the end.

One disadvantage of the second method is the fact that all k rows of the matrix have to be treated, even though a rejection of the value of N may result from one of the first few rows. This early rejection of N is a highly probable event in the case of a quadratic sieve problem. In fact the probability of getting a rejection after h trials is $1 - 2^{-h}$, so that only one N in a thousand is apt to pass the test imposed by the first ten rows. Moreover, in many problems some moduli are much more restrictive than others and if the rows of our matrix which correspond to these moduli are put first, the probability of early rejection is very considerably increased. In order to exploit this possibility we propose another method.

Third Method. This method differs from the previous one in its more rapid treatment of the minor of k rows and s columns. Instead of using the detection of overflow we employ the more elaborate "extract" command. This command, which varies slightly from one binary computer to the next, enables the machine to extract from a given binary number those digits which occupy specified positions. These positions are specified by a number called the extractor. As extract is performed in the SWAC, for example, those digits of a given number, called the "extractee," are selected which correspond to the zero digits of the extractor; all other digits are made zero. Thus if the extractee is

1 1 1 0 1 0 1 0 0 0 1 ···

and the extractor is

$$0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1 \cdots$$

the extracted number is

$$1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0 \cdots.$$

In other words, the extracted number has for its i-th digit the product of the i-th digit of the extractee by the complement of the i-th digit of the extractor.

Let us now consider the minor of k rows and the first s columns of our sieve matrix. Starting with the extractee consisting of s 1's and using the first row of our minor as extractor, the extracted number records by its 1's those columns whose first row elements are zero. Using this extracted number as a new extractee and the second row as extractor, the extracted number now records those columns whose first and second elements are both zero. Continuing this process and comparing the successive extracted numbers with zero (that is the number consisting of s digits 0) after each extraction, we very soon (at least in the typical case) arrive at an extracted number which is zero. At this point the entire minor is thrown away and a new minor is created by circulating the rows of our matrix as described in the second method. If, however, our minor contains one or more columns of zeros, a non-zero number will result from all k extraction operations with digits 1 in the positions corresponding to the solution or solutions of the problem. To find these positions we may proceed successively to double the corresponding number and test for overflow as in Method 1. An obvious method of tallying will produce the solution or solutions which the extraction process has detected.

Comparison of Methods. On the basis of speed, a comparison of the three methods for the case of the quadratic type sieve problem may be made as follows.

Let

T = the average time required to dispose of a single number N and pass on to $N + 1$,
s = the number of binary digits in a machine word,
h = the average number of words for each modulus of the problem,
k = the number of moduli, or width, of the problem,
a = the addition time of the computer, that is, the time required to receive and add two numbers and to dispose of their sum.

For the three methods rough approximations to T may be given as follows:

First Method: $T = 5hka$
Second Method: $T = 5k(1 + 2hs^{-1})a$
Third Method: $T = [10kh + 6(1 + \log_2 s)]s^{-1}a$

As an example of what may be done on the SWAC for which $s = 36$ we take a typical case of $h = 2$ and $k = 17$. The addition time for the SWAC being 64 microseconds, the three values of T turn out to be 10880, 6044, and 672 microseconds respectively.

The above formulas for T do not take into account certain occasional routine operations such as resetting tallies to zero, the restoration of modified commands, the decimal conversion and printing of the solutions, etc. To take these activities into account may lengthen the time by as much as 10 percent. In actual tests on the SWAC, for instance, the third method gave $T = 695$ microseconds. It should be pointed out that h must be at least 2 in the second and third methods since a complete minor of s columns is needed.

Operating Suggestions. We conclude with a few remarks for the benefit of the programmer and operator in dealing with sieve problems. In the normal case the expected output is only a few numbers N occurring in unpredictable places between A and B. It is evident that the machine must be in absolutely perfect operation during the run in order to be sure that no solution has been overlooked. Absolute precision in the circulation of the rows of the sieve matrix is, of course, of paramount importance. The accuracy of this operation may be checked at the end of the run by printing out the entire matrix and inspecting the result. In order to check the machine during operation the programmer may deliberately insert one or more "traps" or precomputed extraneous solutions. It is not difficult to find by hand methods a number N_0 which satisfies most of the requirements imposed by our matrix. To make N_0 satisfy all the conditions we may deliberately weaken the remaining ones by changing a 1 to a 0 in each of the outstanding rows. This weakening, especially when applied to large moduli, will not change perceptibly the overall exclusion ratio but may introduce one or more unexpected extraneous solutions which, however, are easily recognized when they occur. This is a very cheap price to pay for the confidence that one gets from seeing the machine deliver the one or more predicted solutions N_0 on schedule.

In case the running of the problem is interrupted either by machine failure or by another problem of higher priority, there are three procedures available.

First, one may start the problem over from the beginning. This is especially advisable if the interruption occurs early in the run. Secondly, one may make an obvious translation in the variable N and compute by hand a new sieve matrix. Of course, this has to be done with considerable care but in case the machine is being used on another problem, there may be time to do this. In running very long problems it may be wise to prepare in advance several matrices from which a new start may be made in case of interruption. This procedure, however, is almost as much work as breaking up the problem into parts as described above, and this latter procedure always pays off in the consequent speed-up of the problem.

A third procedure is available in case there is still some memory space unused. This method is simply to make the machine compute its new starting matrix itself. This program may be based on the First Method, using detection of overflow to circulate each row of the original matrix to its new position. Of course the amount of this circulation depends on the new proposed starting point and is different for each modulus. If, instead of starting with $N = 1$, we wish to recommence with $N = N_1$, the row corresponding to the modulus m must be circulated $r - 1$ steps where r is the remainder on division of N_1 by m. The program for carrying this out is set

up with N_1 as a free parameter which can be typed into the machine as occasion demands, no further information being needed. This elaboration of the program makes it possible to operate a lengthy sieve problem as a backlog workload without the need of a specially trained operator.

D. H. L.

[1] D. H. LEHMER, "The mechanical combination of linear forms," *Amer. Math. Monthly*, v. 35, 1928, p. 114–121, "A photo-electric number sieve," *Amer. Math. Monthly*, v. 40, 1933, p. 401–406, "A machine for combining sets of linear congruences," *Math. Annalen*, v. 109, 1934, p. 661–667.

[2] L. E. DICKSON, *History of the Theory of Numbers*, v. 2, Washington, 1923; New York, 1934, p. 57.

The Use of Large Intervals in Finite-Difference Equations

In a recent article SIR RICHARD SOUTHWELL[1] has challenged the general theory of finite differences and in particular the use of it in connection with the solution of differential equations by relaxation methods.[2] Opinions differ on the method of treatment of a numerical differentiation formula such as

$$h^2 y_0'' = \left(\delta^2 - \frac{1}{12}\delta^4 + \frac{1}{90}\delta^6 - \cdots\right) y_0,$$

where h is the interval between pivotal points and where $\delta^{2n} y_0$ is the $2n^{\text{th}}$ central difference of y_0. This equation is replaced for the purpose of numerical solution by its equivalent

$$h^2 y_0'' = (y_1 + y_{-1} - 2y_0) + \Delta\,(y_0),$$

where y_1, y_0, and y_{-1} refer to the values of y at $x_0 + h$, x_0, and $x_0 - h$ respectively and

$$\Delta(y) = \left(-\frac{1}{12}\delta^4 + \frac{1}{90}\delta^6 - \cdots\right) y.$$

I advocate the use of as large an interval as possible (consistent with convergence of the finite-difference equations and convenience of computation) calculating $\Delta(y)$ and incorporating it into the computation. Southwell prefers to use such a small interval that Δ is negligible. He states: "*Accuracy is not predictable from quantities computed from a finite-difference approximation unless the interval is small enough to justify belief in the convergence of the Taylor series:* in general the radius of convergence, though it exceeds one or two of the smallest intervals that are practicable, will not exceed many such intervals, and consequently *what have been claimed as closer approximations may in fact be less accurate.*"

The purpose of this note is to point out that, in the writer's opinion, this statement is unduly cautious and that its conclusion cannot be true if finite-difference equations are properly used.

In the first place it is not clear what role the Taylor series plays in finite-difference theory. If the Taylor series is not convergent, care must certainly be exercised; but finite-difference methods do not always break down in this case.[3] The point to be emphasized, however, is that an examination of the differences will tell us, in all practical cases, whether the finite-difference

equations we wish to use are valid and how many differences are significant at the particular interval used. It is stressed by Fox[2] that in all the finite-difference formulae used the differences must be convergent and that otherwise a smaller interval must be used.

By "convergence of differences" I mean that the differences of some order should oscillate about zero, with a maximum amplitude determined in the usual way by the binomial coefficients of that order.[4] The tabulated function can then be regarded as a polynomial, slightly perturbed by rounding errors of not more than half a unit at any point. The degree of the polynomial is the same as the order of the last significant difference, and the function can be interpolated, integrated, and differentiated by finite-difference formulae based on this polynomial representation, the error of the process being of the order of the first neglected term in the finite-difference equation.

The degree of the approximating polynomial varies with the number of figures to which the given tabular entries are rounded off and with the magnitude of the interval between successive pivotal points. The function $(1 + x^2)^{-1}$, for example, can be represented near the origin, at an interval of .1, by a quadratic to two decimals, a sextic to four, and so on. In each case interpolation and integration can be performed effectively to the same precision as that of the tabular entries, though derivatives suffer from the loss of significant figures in successive differences. In particular no estimate can be given for derivatives of the true function of orders greater than the degree of the approximating polynomial.

The fact that the true higher derivatives may be large and ever increasing as for example in the function $(1 + x^2)^{-1}$, is of no significance with regard to the use of finite-difference formulae as applied to the approximating polynomial. If the derivatives increase fast enough such formulae may be asymptotic, but again the error is of the order of the first term neglected. This is analogous to the treatment, in a recent paper by Fox & Miller[3], of a function whose Taylor series had zero radius of convergence at the origin; the series was in fact asymptotic.

Southwell does not use differences and bases his argument on the fact that increased accuracy is not guaranteed by increasing the order of the approximating polynomial, citing as one example the interpolation at unit interval of the function $(1 + x^2)^{-1}$. The first table gives the differences of this function, tabulated to three decimals at unit interval (the central differences at $x = 0$ are filled in from consideration of symmetry). Examination shows that the differences are diverging rapidly, that interpolation formulae can hardly give even one-figure accuracy in the function near $x = 0$, and that a much smaller interval is necessary.

x	10^3y					x	10^4y				
0	1000		−1000		2400	12	6897		285		
		−500		1200				−1015		−76	
1	500		200		−1200	13	5882		209		22
		−300		0				−806		−54	
2	200		200		−141	14	5076		155		16
		−100		−141				−651		−38	
3	100		59			15	4425		117		12
		−41						−534		−26	
4	59					16	3891		91		

The divergence of differences near $x = 0$, however, does not prevent the accurate use of finite-difference formulae at points sufficiently remote from this point. Central-difference formulae, for example, are based on polynomials whose origin is at the particular pivotal point considered, and the function $(1 + x^2)^{-1}$ can be quite correctly interpolated, differentiated, or integrated to at least four figures, at unit interval, in the range shown in the second of the tables.

Turning now to the rest of the quotation, the second part suggests that the largest interval at which the full finite-difference equation can give accuracy will never greatly exceed an interval for which its first term alone is adequate. This will depend, of course, on the particular function involved, but, even if the ratio of intervals is only two, the use of Δ is worth-while not only for economy of effort but also from the point of view of accuracy.

Computers know that the use of small intervals in step-by-step integration is dangerous, the tendency for rounding-off errors to accumulate being roughly proportional to the square root of the number of intervals used. In relaxation methods, similarly, a small interval involves a large number of simultaneous equations, and their accurate solution is a matter of considerable difficulty; often one or even two extra figures have to be retained in the coefficients and residuals. Practical views on this have been reinforced by a recent theoretical paper by TODD.[5]

In illustration of these points let us try to evaluate to four decimals the pathological function $(1 + x^2)^{-1}$ from the facts that it satisfies the differential equation

$$y'' + 4x\,(1 + x^2)^{-1}\,y' + 2\,(1 + x^2)^{-1}\,y = 0,$$

is symmetric about $x = 0$, and has the value $y\,(\tfrac{1}{2}) = .8000$. The finite-difference equation is given by

$$y_1\,(1 + 2x_0 f_0) + y_{-1}\,(1 - 2x_0 f_0) - y_0\,(2 - 2hf_0) + \Delta(y_0) = 0,$$

where

$$\Delta(y) = -\,4xf\left(\frac{1}{6}\,\mu\delta^3 - \frac{1}{30}\,\mu\delta^5 + \cdots\right)y + \left(-\,\frac{1}{12}\,\delta^4 + \frac{1}{90}\,\delta^6 - \cdots\right)y$$

and

$$f = h\,(1 + x^2)^{-1}.$$

At an interval $h = .25$ there are only five pivotal points, giving insufficient information about the differences. The next smallest convenient interval is .1 (assuming we wish to have $x = 0$ as a pivotal point), and at this interval we find the following first approximation $y^{(0)}$, with Δ neglected. Extra figures have been retained in the residuals and coefficients to ensure that this approximation is everywhere an accurate solution (to four figures)

to the given simultaneous equations. The evaluation is as follows:

x	$10^4y^{(0)}$					Δ	R	$10^4y^{(1)}$	Error in $10^4y^{(0)}$
.0	10020		−200		22	−1.8	−1.4	10001	20
		−100		11					
.1	9920		−189		21	−1.9	−2.1	9902	19
		−289		32					
.2	9631		−157		12	−1.5	−1.5	9616	16
		−446		44					
.3	9185		−113		2	−1.0	− .8	9175	11
		−559		46					
.4	8626		−67		(−9)	− .2	− .2	8621	5
		−626		(37)					
.5	8000							8000	0

The differences look quite satisfactory, fifth and higher orders being apparently negligible, and the recorded values of Δ include contributions from third and fourth differences only. If these values are now inserted in the finite-difference equations, the remaining residuals are as shown in the column headed R, and their liquidation gives a better approximation $y^{(1)}$, whose maximum error is one unit. The skilled computer would notice further that the fifth and sixth differences, though small, have a definite flow and are not yet oscillating about zero. Their total contribution is only about −.2, but their effect is to reduce the maximum error in $y^{(1)}$ to half a unit in the last figure.

This work involved the solution of nine simultaneous equations with two sets of residuals and with a little intermediate computation. If Δ were ignored and the computer solved 19 linear equations at an interval of .05, he would still have maximum errors of five units.

It is very doubtful whether intuition alone could here decide the question of the size of the "ultimate interval" for which Δ is negligible, and, if the computer, still ignoring Δ, wished to make sure of his result by taking an interval of .025, he would still have errors of one unit, having solved no less than 39 linear equations in addition to computing the coefficients thereof. The labour of this process is excessive; the accurate solution of these 39 equations would involve keeping at least two extra figures in the coefficients and residuals.

I therefore maintain, with regard to the end of the statement, that a computer familiar with both the theory and practice of finite differences will never be in danger of claiming more accuracy than he has achieved. Either he has got the desired accuracy, or he will know that he has not got it (and perhaps cannot get it, as for example near a discontinuity or singular point). No question of prior knowledge of the convergence of the Taylor series arises, and such knowledge has certainly not been sought in any of my published examples. Examination of the differences, after a tentative solution has been found, provides the necessary information and is of course essential.

In this paper I have concentrated on functions of one variable, but the remarks apply with greater force to partial differential equations, particularly in the desirability, both for the sake of accuracy and a minimum of labour, of using conveniently large intervals.

L. Fox

National Physical Laboratory
Teddington, England
and NBSCL

[1] *Numerical Methods of Analysis in Engineering*. Edited by L. E. Grinter, Macmillan Company, New York, 1949, chapter 4.

[2] L. Fox, "Some improvements in the use of relaxation methods for the solution of ordinary and partial differential equations," Royal Soc., London, *Proc.*, v. 190A, 1947, p. 31–59.

[3] L. Fox & J. C. P. Miller, "Table-making for large arguments. The exponential integral," *MTAC*, v. 4, 1951, p. 163–167.

[4] See, for example, J. C. P. Miller, "Checking by differences—I," *MTAC*, v. 4, 1950, p. 3–11.

[5] J. Todd, "The conditions of a certain matrix," Cambridge Phil. Soc., *Proc.*, v. 46, 1950, p. 116–118.

Monte Carlo Matrix Inversion and Recurrent Events

1. Introduction. Recently Wasow[1] has given a necessary and sufficient condition that one of two unbiased estimators of the inverse element of a given matrix has a smaller variance. Using the theory of recurrent events, we extend a result by Feller[2] in order to generalize and reinterpret Wasow's condition. To do this let us consider a simple discrete Markov process with a finite number of states. Of these $m + 1$ states denoted by $0, 1, 2, \cdots, m$, we prescribe that the state named "0" is the only death state or sink—in the sense that the random walk ends when this state is reached. Let p_{ik} be the one-step transition probabilities for $i, k = 1, \cdots, m$, and let

$$p_{i0} = 1 - \sum_{k=1}^{m} p_{ik} \quad \text{for} \quad i = 1, \cdots, m.$$

Further, we assume that each $p_{i0} > 0$. When $n > 0$ define $p_{ik}^{(n)}$ as the probability that in a random walk starting at state i, state k is visited on the n-th step; and define $p_{ik}^{(0)} = \delta_{ik}$, where δ_{ik} is the Kronecker delta.

Now if $(q_{ik}) = (\delta_{ik} - p_{ik})$ is the m by m matrix whose inverse (q^{ik}) is desired, we may estimate each element q^{ik} by Monte Carlo methods as in Wasow[2], Forsythe & Leibler[3], and Curtiss[4], since

$$q^{ik} = \sum_{n=0}^{\infty} p_{ik}^{(n)}.$$

Two estimators have been investigated. The first estimator is the sample mean of the random variable s_{ik} where $s_{ik} = p_{k0}^{-1}$ if the random walk having started at state i stops just after visiting state k, and $s_{ik} = 0$ if otherwise. The second estimator is the sample mean of the random variable v_{ik} where v_{ik} is the number of visits to state k in a random walk having started at

state i. Now it is known that both these estimators are unbiased, that is,

$$E(s_{ik}) = q^{ik} \tag{1}$$

and

$$E(v_{ik}) = q^{ik} \tag{2}$$

where E is the expected value operator. Further, the second moments of these random variables can be expressed directly in terms of the q^{ik} and p_{k0}. We have

$$E(s_{ik}^2) = p_{k0}^{-1}\, q^{ik} \tag{3}$$

and

$$E(v_{ik}^2) = (2q^{kk} - 1)q^{ik}. \tag{4}$$

Relations (1) and (3) were proved by FORSYTHE & LEIBLER[3], and relations (2) and (4) were derived independently by CURTISS[4] and by the author. Now, Wasow[1] showed that

$$E(s_{ik}^2) = \frac{\lambda_{ik}}{p_{k0}(1 - \lambda_{kk})} \quad \text{for } i \neq k \tag{3'}$$

and

$$E(v_{ik}^2) = \frac{\lambda_{ik}(1 + \lambda_{kk})}{(1 - \lambda_{kk})^2} \quad \text{for } i \neq k \tag{4'}$$

where he defined λ_{ik} as "the probability of going from state i to state k without passing through state k on the way; i.e., λ_{ik} is the total probability associated with all paths connecting state i and state k, all intermediate points being different from state k." Thus, we have before us two different interpretations of the second moments of s_{ik} and v_{ik} in the case $i \neq k$.

2. Recurrent Events. Using the concept of recurrent event as developed by Feller[2] we now extend a result of Feller[2] to establish a general theorem which shows the fundamental probabilistic relation between the two interpretations for the moments. When $n > 0$ define $r_{ik}^{(n)}$ as the probability that in a random walk starting at state i, state k is visited for the first time on the n-th step, and define $r_{ik}^{(0)} = 0$. Let

$$r_{ik} = \sum_{n=0}^{\infty} r_{ik}^{(n)}$$

so that r_{ik} equals the probability that in a random walk starting at state i state k is eventually visited.

THEOREM: $q^{ik} - \delta_{ik} = r_{ik}q^{kk}$.

PROOF: By considering the mutually exclusive and exhaustive cases it follows that

$$p_{ik}^{(n)} = r_{ik}^{(n)} + r_{ik}^{(n-1)}\, p_{kk}^{(1)} + \cdots + r_{ik}^{(1)}\, p_{kk}^{(n-1)} \quad \text{for } n > 0$$

and since $r_{ik}^{(0)} = 0$ and $p_{ik}^{(0)} = \delta_{ik}$ that

$$(*) \qquad p_{ik}^{(n)} = \sum_{j=0}^{n} r_{ik}^{(n-j)} p_{kk}^{(j)} \quad \text{for } n > 0.$$

Next we define

$$P_{ik}(u) = \sum_{n=0}^{\infty} p_{ik}^{(n)} u^n$$

as the probability generating function for the sequence $\{p_{ik}^{(n)}\}$, and define

$$R_{ik}(u) = \sum_{n=0}^{\infty} r_{ik}^{(n)} u^n$$

as the probability generating function for the sequence $\{r_{ik}^{(n)}\}$. Hence by a basic property of generating functions we have by (*)

$$P_{ik}(u) - \delta_{ik} = R_{ik}(u) P_{kk}(u) \quad \text{for all } u.$$

Putting $u = 1$ we get

$$\sum_{n=0}^{\infty} p_{ik}^{(n)} - \delta_{ik} = \left(\sum_{n=0}^{\infty} r_{ik}^{(n)}\right)\left(\sum_{n=0}^{\infty} p_{kk}^{(n)}\right).$$

However, this is

$$q^{ik} - \delta_{ik} = r_{ik} q^{kk}$$

which was to be proved.

As corollaries we get the explicit relations

$$(5) \qquad r_{ik} = \frac{q^{ik} - \delta_{ik}}{q^{kk}}$$

and

$$(6) \qquad q^{ik} = \frac{r_{ik}^{1-\delta_{ik}}}{1 - r_{kk}}$$

since by definition

$$q^{kk} \geq p_{kk}^{(0)} = 1 > 0$$

and by assumption

$$1 - r_{kk} \geq p_{k0} > 0.$$

Thus we have a direct derivation of the general relation between q^{ik} and r_{ik} for all i and k.

As a matter of interest, while working with Wasow, the author first derived relations (5) and (6) indirectly by equating expression (3) to (3′), and expression (4) to (4′). Thus the relation of q^{ik} to λ_{ik} was found. Next it was seen that λ_{ik} is, in fact, just r_{ik}. An informal verification is as follows.

The probability that a random walk initially traverses a given path is equal to the probability that the walk takes that path times the probability that the walk thereafter follows some path from the class of possible paths remaining. But the latter probability is one. Therefore λ_{ik} may be replaced by r_{ik}.

We may now write the second moments of our two random variables in terms of r_{ik}.

$$(7) \qquad E(s_{ik}^2) = \frac{1}{p_{k0}} \frac{r_{ik}^{1-\delta_{ik}}}{1 - r_{kk}}$$

and

$$(8) \qquad E(v_{ik}^2) = \frac{1 + r_{kk}}{1 - r_{kk}} \frac{r_{ik}^{1-\delta_{ik}}}{1 - r_{kk}}.$$

Thus, a comparison of (3) and (4) yields

$$(9) \qquad E(s_{ik}^2) < E(v_{ik}^2) \quad \text{if and only if} \quad \frac{1}{p_{k0}} < 2q^{kk} - 1$$

and similarly, (7) and (8) give

$$(10) \qquad E(s_{ik}^2) < E(v_{ik}^2) \quad \text{if and only if} \quad \frac{1}{p_{k0}} < \frac{1 + r_{kk}}{1 - r_{kk}}.$$

This, of course, is Wasow's result expressed in terms of r_{kk} rather than λ_{kk}. Since r_{kk} is usually unknown, but $r_{kk} \geqslant p_{kk}$; (10) implies

$$(11) \qquad E(s_{ik}^2) < E(v_{ik}^2) \quad \text{if} \quad \frac{1}{p_{k0}} < \frac{1 + p_{kk}}{1 - p_{kk}}.$$

These recurrent events associated with this stochastic process have permitted the derivation of further results which will be treated in a subsequent paper.

H. P. Edmundson

University of California
at Los Angeles

[1] W. R. Wasow, "A note on the inversion of matrices by random walks," *MTAC*, v. 6, 1952, p. 78–81.
[2] W. Feller, *An Introduction to Probability Theory and its Applications*, v. 1, New York, 1950.
[3] G. E. Forsythe & R. A. Leibler, "Matrix inversion by a Monte Carlo method," *MTAC*, v. 4, 1950, p. 127–129.
[4] J. H. Curtiss, "Sampling methods applied to differential and difference equations," IBM *Seminar on Scientific Computations*, November 1949.

RECENT MATHEMATICAL TABLES

1042[A].—C. E. Fröberg, *Hexadecimal Conversion Tables*. Lund, 1952, 20 p., 21.7 × 14.6 cm.

These tables are designed to assist in the coding of problems for binary computers. There are four tables. Table 1 is a table of the integers $1(1)2^{10}(2^4)2^{12}$ in both decimal and hexadecimal notation. Table 2 gives the hexadecimal equivalents of $n \cdot 10^{-2k}$, $n = 1(1)100$, $k = 1(1)8$, correct to 13 hexadecimals. Table 3 gives the hexadecimal equivalent of 50 frequently used constants correct to 16 hexadecimals. Finally Table 4 gives the decimal equivalents of numbers of the form

$$n \cdot 16^{-k} \quad n = 1(1)15, k = 1(1)13.$$

Results are given to 16D. The letters A B C D E F are used as hexadecimal notation for the numbers ten through fifteen. A few examples of the use of the tables are given in the introduction. This handy booklet should find its way into many a coding room.

D. H. L.

1043[F].—M. Kraitchik, *Introduction à la Théorie des Nombres*. Paris, 1952, viii + 202 p., 16.5 × 25.4 cm.

The material for this work is taken largely from the author's *Théorie des Nombres*,[1] v. 1 and 2, and *Recherches*,[2] v. 1. There are numerous small tables which may be reported as follows:

(1) p. 2. Factorization of $\pm 1 + p_1p_2\cdots p_n$, where p_n is the n-th prime, for $n \leq 16$.
(2) p. 4. Factorization of $2^{2^n} + 1$ as far as known in 1947.
(3) p. 5. Factorization of $\pm 1 + n!$ for $n \leq 22$.
(4) p. 12, 13. Complete factorization of $2^n - 1$ for n odd and $n = 1(2)99$, 105, 107, 111(2)117, 123, 127, 135.
(5) p. 38, 39. Factorization of $2^n + 1$ for odd $n \leq 135$ and for even $n \leq 148$ with some gaps.
(6) p. 40, 41. Factorization of $10^n \pm 1$ for $n = 1(2)31$ and $n = 2(2)20$, 24, 30, 36, 50.
(7) p. 55–62. Tables of power residues. These give the least positive or absolutely least residues of k-th powers for prime moduli $< L$ for the following values of k and L

$k =$	2,	3,	4,	5,	6,	7,	11
$L =$	200,	200,	200,	422,	242,	632,	1000

(8) p. 124–129. Tables of linear divisors of $x^2 - Dy^2$. These tables are in three parts. The first part gives the arithmetical progressions in which p must lie if D is to be a quadratic residue of p for all square free numbers D less than 35 in absolute value. The second part gives this information for $0 < D = 4k + 1 \leq 241$. The third part is for $0 > D = 4k + 3 \geq -239$. These contain two errata, carried over from the corresponding tables in *Théorie des Nombres*, v. 1:

p. 125 $D = 157$ *for* 107 *read* 109
p. 126 $D = 193$ *for* 155 *read* 129.

(9) p. 165–177. Tables of $x \pmod p$ in $x^2 - y^2 = N$. These are for all primes $p \leq 67$ and are taken from *Théorie des Nombres*, v. 2, p. 156–166.
(10) p. 182. Table of a cyclotomic quadratic form. This table gives the coefficients of the polynomials X, Y, such that

$$X_n^2(x) - nxY_n^2(x) = \begin{cases} Q_n(x) & n = 4k+1 \\ Q_{2n}(x) & n = 4k+3 \end{cases}$$

for all square free $n \leq 35$ and also 39, 42, and 51. It is taken from *Recherches*,[2] v. 1, p. 88, with two errata. The coefficients of X_{29} should be 1, 15, 33, 13, 15, $\cdots$, 15, 13, 33, 15, 1.
(11) p. 184–186. Factorization of $2^{4n+2} + 1$. The table extends to all $4n + 2 \leq 162$ as well as 170, 174, 182, 186, 190, 198, 210, 222, 234, 250, 258, 270, and 330, with some results partially incomplete. For $n = 94$, 114, and 150, results are put in the wrong columns of the table.

D. H. L.

[1] M. Kraitchik, *Théorie des Nombres*. v. 1, Paris, 1922, ix + 229 p., v. 2, Paris, 1926.
[2] M. Kraitchik, *Recherches sur la Théorie des Nombres*. v. 1, Paris, 1924.

1044[F].—A. NATUCCI, "Osservazioni sul problema di Fermat," Un. Mat. Ital., *Boll.*, s. 3, v. 6, 1951, p. 245–248

This paper contains a short table of

$$F_n(x, u, v) = (x + u + v)^n - \{x^n + (x + u)^n\}$$

for $v = 1$, $u = 1, 2$ and $n = 3, 4, 5$. The variable x ranges over integers until F becomes negative. Two or three negative values are given and x does not exceed 13.

D. H. L.

1045[F].—GIUSEPPE PALAMÀ, "Numeri primi e composti contenuti nella forma $1848x^2 + y^2$ dell' intervallo 11000000 − 11100000," Unione Mat. Ital., *Boll.*, s. 3, v. 7, p. 168–171, 1952.

The author uses a recent list of primes of the eleventh million[1] to find all the primes of the form $1848x^2 + y^2$ between the above limits together with values x and y. There are 202 primes. This is similar to a list above 10 million given by CUNNINGHAM & CULLEN.[2] The author also lists 121 composite numbers of this form.

D. H. L.

[1] J. P. KULIK, L. POLETTI & R. J. PORTER, *Liste des nombres premiers du onzième million*. Amsterdam, 1951 [*MTAC*, v. 6, p. 81, errata p. 163].

[2] J. C. CUNNINGHAM & J. CULLEN, "On idoneal primes," Br. Asso. Adv. Sci., *Report*, 1901, p. 552.

1046[H].—P. T. SMOLÎAKOV & A. N. HOVANSKIĬ, "K resheniu algebraicheskikh uravneniĭ 3–ĭ stepeni" [On the solution of algebraic equations of the third degree], Kazan Filial. Akad. Nauk. SSSR., *Izvestiîa* Fiz-Mat. Tekhn. Nauk, v. 1, 1948, p. 85–92.

The authors take as a reduced form the cubic

$$y^3 + Py + 1 = 0.$$

On p. 87 is a 2D table giving one real root y corresponding to 271 given values of P between − 10 and 10.

D. H. L.

1047[K].—G. E. ALBERT & R. B. JOHNSON, "On the estimation of central intervals which contain assigned proportions of a normal univariate population," *Annals Math. Stat.*, v. 22, 1951, p. 596–599.

Let Y be a random variable with the cumulative probability function $F(y) = (\sigma^2 2\pi)^{-\frac{1}{2}} \int_{-\infty}^{y} \exp[-(u - m)^2/2\sigma^2]du$. The statistic considered is

$$A(\bar{Y}, s; \lambda) = F(\bar{Y} + \lambda s) - F(\bar{Y} - \lambda s)$$

where $\bar{Y} = \sum_{i=1}^{N} Y_i/N$, $s = \left\{\sum_{i=1}^{N} (Y_i - \bar{Y})^2/(N - 1)\right\}^{\frac{1}{2}}$.

WILKS has shown[1] that if for given p, $0 < p < 1$, one sets

$$\lambda_{p,N} = t_p[(N + 1)/N]^{\frac{1}{2}},$$

where t_p is Student's t, then the expectation of this statistic is

$$E[A(\bar{Y}, s; \lambda_{p,N})] = 1 - p.$$

The problem treated in the present paper is: for given p, α, d_1, d_2, such that $0 < p < 1$, $0 < \alpha < 1$, $0 < d_1 \leqq 1 - p$; $0 < d_2 \leqq p$, find the smallest sample size N for which one has

$$\text{Prob. } \{1 - p - d_1 \leqq A(\bar{Y}, s; \lambda_{p,N}) \leqq 1 - p + d_2\} \geqq \alpha.$$

A table of such smallest sample values is given for $p = .01$, $\alpha = .95, .99$, and for $p = .05, .25, .50$, $\alpha = .80, .95, .99$, in each case for $(d_1, d_2) = (.075, .05)$, $(.05, .05)$, $(.025, .025)$, $(.035, .015)$, $(.05, .01)$, $(.025, .01)$, $(.02, .01)$, $(.01, .01)$.

Z. W. BIRNBAUM

Univ. of Washington
Seattle, Wash.

[1] S. S. WILKS, "On the determination of sample sizes for setting tolerance limits," *Annals Math. Stat.*, v. 12, 1941, p. 91–96.

1048[K].—D. H. BHATE, "A note on the significance levels of the distribution of the mean of a rectangular population," Calcutta Stat. Assn., *Bulletin*, v. 3, 1951, p. 172–173.

For means of samples of n drawn from a continuous rectangular universe on the interval (0, 1) the author gives the 5%, 2.5% and .5% points to 4D for $n = 1(1)16$. He remarks that for $n \geqq 16$ the corresponding percentage points for the normal distribution function approximation are correct to 4D.

C. C. C.

1049[K].—J. M. CAMERON, "Tables for constructing and for computing the operating characteristics of single-sampling plans," *Industrial Quality Control*, v. 9, 1952, no. 1, p. 37–39.

Let the probability of acceptance of a lot be denoted by $P(A)$ and the lot fraction defective by p. For a single-sampling plan with sample size n and allowable number of defectives c, Table 1 provides 13 paired values of p and $P(A)$ which can be used to plot the respective operating characteristic (OC) curves. The entries in Table 1 are values of np which are given to 3D for $c = 0(1)49$ and for $P(A) = .995, .990, .975, .950, .900, .750, .500, .250, .100, .050, .025, .010, .005$.

Table 2 is designed for constructing single-sampling plans whose OC curve passes through two points $(p_1, 1 - \alpha)$ and (p_2, β) where p_1 is the fraction defective for which the risk of rejection is to be α, and where p_2 is the fraction defective for which the risk of acceptance is to be β. The entries in Table 2 are values, given to 3D, of the ratio p_2/p_1 which are associated with three sets of paired values of α and β; namely, $(\alpha = .05, \beta = .10)$, $(\alpha = .05, \beta = .05)$, and $(\alpha = .05, \beta = .01)$. Corresponding to the selected value of p_2/p_1 are values of np_1 and c. The sample size is determined by dividing np_1 by p_1 and the acceptance number is read directly from the table.

Since the tables were computed from the Poisson series, they are exact only when the probability distribution of the number of defectives in a sample of n follows the Poisson law. However, for most practical cases, the

tables will give satisfactory approximations when the distribution of defectives is binomial.

G. W. McElrath

University of Minnesota
Minneapolis, Minnesota

1050[K].—J. H. Chung & D. B. DeLury. *Confidence Limits for the Hypergeometric Distribution.* Univ. of Toronto Press, Toronto 1950, 72 leaves, 22.5 × 30.9 cm. Price $2.25.

The charts in this book for the hypergeometric distribution are similar to the ones for the binomial distribution given by Clopper & Pearson.[1] Charts are given for population sizes 500, 2500, and 10,000, for sampling rates 5%, 10% (10%) 90%, and for confidence coefficients 90%, 95%, and 99%. The methods of constructing the charts are given explicitly, as well as means for performing interpolation and extrapolation to cases not included.

I. R. Savage

National Bureau of Standards
Statistical Engineering Laboratory
Washington, D. C.

[1] C. J. Clopper & E. S. Pearson, "The use of confidence and fiducial limits applied to the case of the binomial," *Biometrika*, v. 26, 1934, p. 404–413.

1051[K].—A. Hald & S. A. Sinkbaek, "A table of percentage points of the χ^2-distribution," *Skandinavisk Aktuarietidskrift*, v. 33, 1950, p. 168–175.

The present table gives values of χ_0^2 to 3D for which $P(\chi^2 \geqq \chi_0^2)$ in which χ^2 has f degrees of freedom for $P = .0005, .001, .005, .01, .025, .05, .1(.1).9, .95, .975, .99, .995, .999, .9995$ and $f = 1(1)100$.

C. C. C.

1052[K].—H. Levene, "On the power function of tests of randomness based on runs up and down," *Annals Math. Stat.*, v. 23, 1952, p. 34–56.

In the set of observed values, $x_1, x_2, \cdots, x_n$ of a continuous statistical variable X, let B^* be the sequence of signs ($+$ or $-$) of the differences $(x_{i+1} - x_i)$ for $i = 1, \cdots, n-1$. A sequence of successive $+$ $(-)$ signs not immediately preceded or followed by a $+$ $(-)$ sign is called a run up (down). Let s be the number of runs up and let $E'(s) = \lim_{n\to\infty} \frac{1}{n} E(s)$, $\sigma'^2(s) = \lim_{n\to\infty} \frac{1}{n} \sigma^2(s)$, where $E(s)$ is the expected value of s and $\sigma^2(s)$ is the variance of s. Also let k be the total number of $+$ signs in B^*, with $E'(k)$ and $\sigma'^2(k)$ defined analogously to $E'(s)$ and $\sigma'^2(s)$. s and k are members of a class of u-run statistics which may be used for tests of randomness of the sample values, other members of which are also discussed by the author.

One of the alternatives to randomness, precisely defined in this paper, is that x_i obeys a normal distribution law with mean $i\vartheta$ and unit variance, $i = 1, \cdots, n$. For this case $E'(s)$, $1 - E'(k)$, $\sigma'^2(k)$, and $\sigma'(k)$ are given to 6D for the first three and to 3D for the fourth for $\vartheta/\sqrt{2} = 0(.1)3(.2)4$ in Table 1. An interesting graph of $E'(s)$ for the linear trend considered compares the results for normal and rectangular universes.

L. A. Aroian

Hughes Aircraft Company
Culver City, California

1053[K].—F. H. C. MARRIOTT, "Tests of significance in canonical analysis," *Biometrika*, v. 39, 1952, p. 58–64.

Given two sets of variates $x_1, \cdots, x_p$ and $y_1, \cdots, y_q$ ($p \leqq q$) and $n + 1$ observations on each set, the canonical correlations are defined as the p roots ($l_1^2 \geqq l_2^2 \geqq \cdots \geqq l_p^2$) of the determinantal equation $|Q - l^2T| = 0$, where T is the dispersion matrix of the y's; W is the dispersion matrix of the y's with the x's eliminated; $Q = T - W$ is the dispersion matrix due to regression.

The exact null distribution of the largest root l_1^2 is given by ROY[1] in a recursive form involving multiple integrals. The author gives the exact distribution of l_1^2 for $p = 2$ and $p = 3$, $q = 4$. A significance test which is exact for practical purposes is given for $p = 3$ and $p = 4$, $q = 5$. 5% and 1% significance levels to 2D (3 or 4S) for $\frac{1}{2}n\, l_1^2$ (n large) are given for $p = 2$, $q = 2(1)12, 21$; $p = 3$, $q = 3(1)12, 21$; $p = 4$, $q = 5$.

An approximate test,

$$\chi^2_{[D]} = -\ \{n - \tfrac{1}{2}(p + q + 1)\} \log (1 - l_1^2)$$

where $D = p + q - 1 + \frac{1}{2}\{(p - 1)\,(q - 1)\}^{\frac{1}{2}}$, based on WILKS' criterion[2] is proposed. 5% and 1% significance levels to 2D (3 or 4S) for $\frac{1}{2}n\, l_1^2$ (n large) are given for $p = 2$, $q = 2, 6, 12, 21$; $p = 3$, $q = 3, 6, 12, 21$; $p = 4$, $q = 5$ which compare favorably with the exact values. The derivation of the test is omitted, with reference to the author's unpublished thesis.[3] The 5% points of l_1^2 for the exact and approximate tests for $p = 2$, $q = 5$ are given to 2 or 3D for $n = 10, 20, 50, 100, \infty$, which indicate good results for n down to 20.

I. OLKIN

Michigan State College
East Lansing, Michigan

[1] S. N. ROY, "The individual sampling distribution of the maximum, the minimum, and any intermediate of the p-statistics on the null hypothesis," *Sankhyā*, v. 7, 1945, p. 133–158.

[2] S. S. WILKS, "Certain generalizations in the analysis of variance," *Biometrika*, v. 24, 1932, p. 471–494.

[3] F. H. C. MARRIOTT, *The Analysis and Interpretation of Multiple Measurements*. 1951, University of Aberdeen.

1054[K].—SIGEITI MORIGUTI, "A lower bound for a probability moment of any absolutely continuous distribution with finite variance," *Annals Math. Stat.*, v. 23, 1952, p. 286–289.

The n-th probability moment of a population with probability density function $f(x)$ is defined as

$$\Omega_n = \int_{-\infty}^{\infty} [f(x)]^n\, dx.$$

The author considers populations with zero mean and finite variance σ^2. A table is computed for the lower bound of the reduced probability moment $\Omega_n\sigma^{n-1}$ for $n = 2(1)10$ to 5D and a population which attains the lower bound is exhibited. The problem arose in the distribution of sample ranges.

J. R. VATNSDAL

State College of Washington
Pullman, Washington

1055[K].—Seiji Nabeya, "Absolute moments in 2-dimensional normal distribution," Inst. Stat. Math., *Annals*, v. 3, 1951, p. 2–6.

Let x and y have the joint probability density function

$$\frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}\left(\frac{x^2}{\sigma_1^2} - \frac{2\rho xy}{\sigma_1\sigma_2} + \frac{y^2}{\sigma_2^2}\right)\right].$$

This paper presents expressions for $E(|x^m y^n|)$ which are functions of σ_1, σ_2, and ρ. The cases considered are those for which simultaneously $m \geqq n$, $m \geqq 1$, $n \geqq 0$, $m + n \leqq 12$.

J. E. Walsh

U. S. Naval Ordnance Test Station
China Lake, California

1056[K].—K. R. Nair, "Some three-replicate partially balanced designs," Calcutta Stat. Assn., *Bulletin*, v. 4, 1951, p. 39–42.

Four types of two-associate and four types of three-associate designs, using three replicates of each treatment, are discussed. A list of 48 useful designs is presented for which v (number of treatments) $\geqq 10$ and k (number of treatments per block) $\leqq 10$. The number of associate classes (classes of treatment comparisons) and the efficiency factor are given for each design.

R. L. Anderson

North Carolina State College
Raleigh, N. C.

1057[K].—M. H. Quenouille, "The variate-difference method in theory and practice," International Stat. Inst., *Review*, v. 19, 1951, p. 121–129.

The author discusses the usual use of the variate difference method to estimate the random variance in a time series and to test for the existence of trend after the i-th difference. He develops a new test of the existence of trend and a formula to estimate the random variance based on differences of the variances of the successive variate differences. If we let V_i be the variance of the i-th set of variate differences (as defined by the author), then the variance of $\Delta^m V_i$ is $\alpha_{mi}\sigma^4/n$, where n is the number of observations, σ^2 is the true random variance, and values of α_{mi} are given by the author for $i = 1(1)10$ and $m = 0(1)5$, $m + i \leqq 10$, to 4D for $m = 0$, 1 and to 6D for $m \geqq 2$.

The author often interchanges his notation (e.g., m is an order of differencing and also indicates both a terminal and an intermediate variance, V_m). He does not indicate how to decide on the best of many estimates to use for the random variance. However, his methods may be quite useful, if a systematic procedure is developed for using them.

Some discussion is made of the difficulties involved when the residuals are serially correlated. The author discusses two empirical examples not involving serial correlation and one practical example with serially correlated residuals. More discussion is needed of how the empirical data were generated.

R. L. Anderson

North Carolina State College
Raleigh, N. C.

1058[K].—P. V. Sukhatme, V. D. Thawani, V. G. Pendharkar, & N. P. Natu, "Revised tables for the d-test of significance," Indian Soc. of Agricultural Stat., *Jn.*, v. 3, 1951, p. 9–23.

The d-test under discussion is the Behrens-Fisher test for the difference of two sample means drawn from normal universes with possibly unequal variances. Specifically,

$$d = (\bar{x}_2 - \bar{x}_1)/(s_2^2 + s_1^2)^{\frac{1}{2}} = t_2 \cos \vartheta - t_1 \sin \vartheta$$

in which $\bar{x}_1$ and $\bar{x}_2$ are the sample means, s_1^2 and s_2^2 are unbiased estimates of the population variances, $\tan \vartheta = s_1/s_2$ and t_1 and t_2 are Student-Fisher t's with n_1 and n_2 degrees of freedom. Sukhatme's original tables of 5% and 1% values of d appeared in the 1943 edition of Fisher & Yates' tables.[1] Meanwhile Fisher developed an asymptotic method of calculating these percentage points[2] and detected a small positive bias in Sukhatme's values. The latter investigated further and found that his errors were due to his use of linear interpolation in Student's original tables of the t-distribution.[3] His corrected 5% values were used in the 1948 edition of the Fisher & Yates' tables but his 1% values were corrected by Fisher for a positive bias shown in comparison with those of Fisher. In the present paper the authors report the results of a further investigation. They find that the positive bias reported by Fisher did not exceed unity in the third decimal place and that the use of quadratic interpolation is sufficient to give accuracy to 3D for the arguments used. Their investigation of Fisher's method casts doubt on its adequacy at all points. Their finally corrected values appear as their Tables III and VII. These are for the same arguments as those in the Fisher & Yates tables and are also to 3D. (Values are apparently given to 4D when the final figure is 5 but no indication is given whether such a 5 is 5^+ or 5^-.) The 5% values agree with the 1948 Fisher & Yates values but there are three discrepancies in the 1% values as follows:

n_1	n_2	ϑ	Sukhatme	Fisher & Yates
24	24	15°(75°)	2.784	2.785
6	6	30°(60°)	3.558	3.557
8	8	30°(60°)	3.241	3.239

C. C. C.

[1] R. A. Fisher & F. Yates, *Statistical Tables for Biological, Agricultural and Medical Research.* London. 2nd ed. 1943, 3rd ed. 1948. Table V_1.

[2] R. A. Fisher, "The asymptotic approach to Behrens's integral, with further tables for the d-test of significance," *Ann. Eugenics*, v. 11, 1941, p. 141–172.

[3] Student, "New tables for testing the significance of observations," *Metron*, v. 5, 1925, p. 105–120.

1059[L].—D. Caligo, "Complementi analitici e numerici allo studio delle aste vibranti," Accad. naz. dei Lincei, Rome, *cl. sci. fis. mat. nat.*, *Rendiconti*, s. 7, v. 12, 1952, p. 76–83, 277–285.

Let β_ρ, $\rho = 0, 1, \cdots$ be the roots of $\cosh x \cos x = 1$, $\rho = 0$ and $\rho = 1$ being the double root. Table I, p. 82, gives values of $\beta_\rho, (-1)^\rho[\beta_\rho - (\rho - \frac{1}{2})\pi]$, and $\sinh \beta_\rho/[\cosh \beta_\rho + (-1)^\rho]$ for $\rho = 2(1)5$.

Table V, p. 282, contains similar information for the roots of

$$\cosh x \cos x = -1.$$

Six other tables give values of certain integrals occurring in the physical problem.

A. E.

1060[L].—M. A. DENGLER, M. GOLAND & Y. L. LUKE, *Tables of the functions* $\int_0^y (e^{iu} - 1) \sqrt{u^2 + b^2}\, du/u$. Midwest Research Institute, *Report* August 1952, 19 p.

This report tabulates

$$A(y, b) = \int_0^y u^{-1}(1 - \cos u)\, (u^2 + b^2)^{\frac{1}{2}}\, du$$

$$B(y, b) = \int_0^y u^{-1} \sin u\, (u^2 + b^2)^{\frac{1}{2}}\, du$$

for

$$b = 0(.2)1.8, \quad y = 0(.05)2(.1)2.5$$

and

$$b = 2(.2)4, \quad y = 0(.05).5(.1)2.$$

The introduction describes the computation. 9D were carried, and the results have been rounded to 7D for presentation in these tables. The entries should be correct to within one unit in the seventh place. Five-point Lagrangean interpolation in the y direction gives results correct to about two units in the sixth place. If $b > 1$, similar interpolation in the b-direction is reliable to about one unit in the fifth place. If $b < 1$, interpolation in the b direction is poor.

A. E.

1061[L].—C. FERRARI, "The turbulent boundary layer in a compressible fluid with positive pressure gradient," *Jn. Aero. Sci.*, v. 18, 1951, p. 460–477.

There is a table (p. 476) of $G_n(x) = 2^{2n} n!\, i^{2n} \operatorname{erfc} x$ for $x = 0(.05)1(.1)2.2$, $n = 2(1)5$. The values are to 4D but "the fourth place is not accurate." This table extends one by HARTREE.[1]

D. H. L.

[1] D. R. HARTREE, "Tables of the error function," Manchester Lit. and Phil. Soc. *Memoirs and Proceedings*, v. 80, 1935, p. 85. [Reprinted in CARSLAW & JAEGER, *Conduction of Heat in Solids*, p. 373, Appendix III.]

1062[L].—H. J. GODWIN, "A method for the evaluation of

$$\int_0^\infty x^m \left(\sqrt{(2/\pi)} \int_x^\infty \exp\,(-\tfrac{1}{2}t^2)\, dt \right)^n dx,"$$

Quart. Jn. Mech. Appl. Math., v. 5, 1952, p. 109–115.

Put

$$(2/\pi)^{\frac{1}{2}} \int_x^\infty \exp(-\tfrac{1}{2}t^2)\, dt = g(x),$$

$$\int_0^\infty x^m \{g(x)\}^n\, dx = I_{mn},$$

$$P(m, 0, x) = x^m,$$

$$P(m, r, x) = rxP(m, r-1, x) + P'(m, r-1, x),$$

$$\gamma_{m,v} = P(m, m+2v, 0)\, (\tfrac{1}{2}\pi)^{\frac{1}{2}(m+2v+1)}/(m+2v)\,!$$

The author computes $\gamma_{m,v}$ and I_{mn} by expansion in inverse factorial series. He gives 9–10D tables of I_{mn} for $m = 0, 1, 2$, $n = 1(1)20$, and 9–15D tables of γ_{mv} for $m = 0$, $v = 0(1)18$; $m = 1, 2$, $v = 0(1)10$.

A. E.

1063[L].—V. R. THIRUVENKATACHAR & B. S. RAMAKRISHNA, "A case of combined radial and axial heat flow in composite cylinders," *Quart. Appl. Math.*, v. 10, 1952, p. 255–262.

On p. 261 there is a 2D table for the first four roots of the equation

$$10x\, \frac{J_1(x)}{J_0(x)} = \frac{J_1(3y/2)\, Y_1(y) - J_1(y)\, Y_1(3y/2)}{J_1(3y/2)\, Y_0(y) - J_0(y)\, Y_1(3y/2)}\, y$$

where

$$y = [10x^2 + .09\pi^2 n^2]^{\frac{1}{2}}, \quad n = 1, 3, 5.$$

A. E.

1064[V].—ROBERT W. SMITH, JR., HELEN E. EDWARDS & STUART R. BRINKLEY, JR., *Tables of Velocity of Steady Laminar Flow in Channels of Rectangular Cross Section.* Report of Investigations 4885, U. S. Dept. of the Interior, Bureau of Mines, July 1952, 41 p.

Table 2 contains the solution of the boundary value problem: $\omega_{xx} + \omega_{\bar{y}\bar{y}} = -2$ in the rectangle $R(|x| \leq 1, |\bar{y}| \leq 1/\sigma)$, $\omega = 0$ on the boundary of R. For each $\sigma = .1(.1)1$, $\omega(x, y)$ is presented as a function of $x = 0(.1).5(.05).7(.02)1$ and $y = \sigma\bar{y}$. In the first 20 columns $y = 0(.1).5(.05).7(.02).9$; in the remaining columns $y = .905(.005)1$ for $\sigma = .1$ and $.2$, $y = .91(.01)1$ for $\sigma = .3(.1).7$, and $y = .92(.02)1$ for $\sigma = .8(.1)1$.

For the same range of σ Table 1 has $\omega_0 \equiv \omega(0, 0)$, $\bar{\omega} \equiv \int_0^1 \int_0^1 \omega(x, y) \times dxdy$, $k' \equiv \sigma\omega_0/8$, $k \equiv \sigma\bar{\omega}/8$; Table 3 has $(-\partial\omega/\partial x)_{x=1}$ for the above pairs of values of σ and y; and Table 4 has $(-\partial\omega/\partial y)_{y=1}$ for the above values of x.

In hydrodynamics ω is the reduced velocity of the steady laminar flow of a viscous fluid through a channel with the rectangular cross section R.

In the problem of twisting an isotropic prismatic bar with cross section R and shear modulus μ by a couple M whose moment is directed along the axis of the bar, ω is the stress function. Moreover, if α is the angular twist per unit length of the bar, $M = 8\mu\alpha\bar{\omega}/\sigma$ and the stress components are $\tau_{xz} = \sigma\mu\alpha\partial\omega/\partial y$ and $\tau_{zy} = -\mu\alpha\partial\omega/\partial x$.[1]

The tabulated functions were computed to 10D from Fourier series and rounded off to 6D.

The very last factor in equations (17) should be $[g(0, y, \sigma)]^2$. The second of equations (12) is $k' = \cdots$. The denominators of the factor before $\sum$ in equations (5), (7), (9), and (14) are π^3, π^3, π^5, and π^2.

R. R. REYNOLDS

NBSINA

[1] I. S. SOKOLNIKOFF, *Mathematical Theory of Elasticity*. New York, McGraw-Hill, 1946. See under "Torsion" in index for references to electrical, hydrodynamic, and membrane analogies. The right side of equation (38.15), p. 149, of the above reference is the series for $\omega + (x^2 + \bar{y}^2)/2$ provided y, a, and b are replaced by $\bar{y}$, 2, and $2/\sigma$.

MATHEMATICAL TABLES—ERRATA

In this issue references have been made to RMT **1043**, **1058**, **1061**, and **1064**.

218.—N. W. McLACHLAN & P. HUMBERT, *Formulaire pour le Calcul Symbolique*. Mémorial des Sciences Mathématiques, fasc. 100, 2nd ed., Paris 1950.

Errata in the first edition have been noted in MTE **205** [*MTAC*, v. 6, p. 100-101]. Beside these errata the second edition contains also the following:

p. 4, formula 5; *for* $t^2/4$ *read* $t^2/2$
p. 4, formula 9; *for* E *read* F
p. 6, formula 8; *for* t^a *read* t^{-a}
p. 11, formula 2; *for* $\begin{matrix} p \\ a \end{matrix}$ *read* $\dfrac{p}{a}$
p. 12, formula 7; *for* $-\phi'(-\log p)$ *read* $\phi(\log p) - f(0)$
p. 12, formula 9; *for* 2^n *read* $2^{n/2}$ and *for* $x/2\sqrt{t}$ *read* $x/\sqrt{2t}$
p. 16, formula 9; *add* $R(\nu) > 1$
p. 21, formulas 6, 7; *add* $0 < t < 1$
p. 21, formula 11; *add* $t > a$
p. 22, formula 10; *for* $\mathrm{ch}^{2n}t$ *read* $\mathrm{sh}^{2n+1}t$ (see *Supplément*, p. 10)
p. 22, formula 11; *for* $\mathrm{sh}^{2n+1}t$ *read* $\mathrm{sh}\sqrt{t}$
p. 22, last formula; *delete* the brackets [ ]
p. 23, formula 11; *for* $p \log (p/\sqrt{p^2 - 1})$ *read* $p \log (\sqrt{p^2 - 1}/p)$
p. 23, formula 12; *for* $p \log \left(\dfrac{\sqrt{p^2 - 1}}{p} \right)$ *read* $p^2 \log \left(\dfrac{\sqrt{p^2 - 1}}{p} \right)$
p. 25, formula 6; the l.h.s. is equal to $\nu^{-1}\,\mathrm{sh}(\nu \operatorname{arg\,ch} t)$
p. 27, formula 6; *for* $1/(p + 1)$ *read* $p/(p - 1)$
p. 27, last formula but one; first term, *for* numerator on r.h.s., *read* a
p. 28, formula 6; *for* $R(\nu) > -\frac{1}{2}$ *read* $R(\nu) > -1$
p. 28, third last formula; *for* 2^ν *read* 2^μ and *for* $R(\mu + \nu) > 0$ *read* $R(\mu + \nu) > -1$
p. 28, last formula; *for* $J_\nu(t)$ *read* $J_{2\nu}(t)$, *for* $(2\nu q - p)$ *read* $(2\nu q + p)$ and *for* $R(\nu) > 1$ *read* $R(\nu) > \frac{1}{2}$
p. 29, formula 5; *add* $R(\nu) > -2$
p. 29, formula 9; multiply the r.h.s. by $n!$ (see *Supplément* p. 11)

p. 30, third last formula; in numerator *read* $(-1)^m a^{\nu+2m} b^{\frac{1}{2}(\nu+1)+m} \times \Gamma\left(\frac{\nu}{2}+m+1\right)$ and *add* $R(\nu) > -2$
p. 31, formula 5; *for* J_ν *read* J_ν^2
p. 35, third last formula; *for* $R(\mu+\nu) > 1$ *read* $R(\mu+\nu) > -1$
p. 35, last formula; *for* I_ν *read* $I_{2\nu}$, *for* $(2\nu u - p)$ *read* $(2\nu u + p)$, and *for* $R(\nu) > 1$ *read* $R(\nu) > \frac{1}{2}$
p. 36, formula 3; *delete*
p. 36, formula 4; on r.h.s. *alter* center sign to + and *add* $R(\nu) > -2$
p. 36, formula 10; *for* $\sum_{r=1}^{\infty} I_{\nu+r}(2/p)$ *read* $\sum_{r=1}^{\infty} (-1)^r I_{\nu+r}(2/p)$
p. 36, last two formulae; *delete*
p. 37, formula 9; *replace* r.h.s. by $p(\sqrt{p+a}-\sqrt{p})^{2\nu}/\nu a^\nu$, and *add* $R(\nu) > 0$
p. 37, formula 10; *for* e^{-at} *read* e^{-at}
p. 37, formula 12; *for* ap/s *read* ap/s^2
p. 39, formula 9; *for* $\frac{1}{4} > R(\nu) > -\frac{1}{4}$ *read* $-1 < R(\nu) < 1$
p. 39, formula 11; to l.h.s. *add* $-\frac{1}{2}\log\{(t-b)/(t+b)\}\, I_0(ay)$
p. 39, formula 12; to l.h.s. *add* $-\frac{t}{2}\log\{(t-b)/(t+b)\}\, I_0(ay)$ and *for* p in [ ] *read* $-p$
p. 39, formula 13; *delete*
p. 39, formula 14; for l.h.s. *read*

$$\left(\frac{t-b}{t+b}\right)^{\nu/2}\left[K_\nu(ay) - \frac{\pi}{2\sin\nu\pi} I_{-\nu}(ay)\left\{1-\left(\frac{t+b}{t-b}\right)^\nu\right\}\right]$$

p. 40 fourth last formula: *for* $-2\sqrt{t}$ *read* $2\sqrt{it}$ and *for* $(-)^{n/2}$ *read* $e^{in\pi/2}$
p. 42, formula 3; to l.h.s. *add* $-\frac{1}{2}\log\{(t-b)/(t+b)\}$ (ber $ay + i$ bei ay)
p. 42, formula 4; *for* the l.h.s. *read*

$$\left(\frac{t-b}{t+b}\right)^{\nu/2}\Bigg\{(\mathrm{ker}_\nu\, ay + i\,\mathrm{kei}_\nu\, ay) - \frac{\pi}{2\sin\nu\pi}(\mathrm{ber}_{-\nu}\, ay + i\,\mathrm{bei}_{-\nu}\, ay)\left[1-\left(\frac{t+b}{t-b}\right)^\nu\right]\Bigg\}$$

p. 42, formula 5; to l.h.s. *add* $-\frac{t}{2}\log\left(\frac{t-b}{t+b}\right)$ (ber $ay + i$ bei ay)
p. 42, formula 6; *delete*
p. 47, formula 4; *for* $-a^2/s^2$ *read* $-a^2/s^3$
p. 48, last formula; *for* p^{2n+1} *read* p^{2n}
p. 50, third last formula; *for* $t^{-4/5}$ *read* $t^{-5/4}$
p. 51, replace this page by p. 53 of the *Supplément*
p. 55, formula 3; *for* 2^n *read* $2^{n/2}$ and *for* $a/2\sqrt{t}$ *read* $a/\sqrt{2t}$
p. 55, fifth last formula; *replace* r.h.s. by $p[(1-1/p)^m - 1]$
p. 56, formula 13; *for* L_{2n} *read* L_{2n}^a
p. 57, formulas 1, 2; *delete*
p. 57, last formula; *for* $[1+\beta p]^{n+a-1}$ *read* $[1+\beta p]^{n+a+1}$
p. 58, formula 6; *for* + arc tg p *read* − arc tg p

p. 58, last formula but one; *for* ϵ^{-ap} *read* ϵ^{-ap}
p. 59, formula 1; *for* $(1 + h/p)$ *read* $(1 + (hp)^{-1})$
p. 59, third last formula; *for* $2A_1$ *read* A_1

I am indebted to A. ERDÉLYI for many of these corrections, some of which were communicated to him by O. VOELKER.

N. W. MCLACHLAN

Vizand & Co.
51 Lincoln's Inn Fields
London W.C.2., England

219.—NBSMTP., *Tables of Fractional Powers*. New York, 1946.
Table 3, p. 34, *for* $\pi^{-10} = 1.0678289226\cdots$
read $\pi^{-10} = 1.0678279226\cdots$.

MURLAN S. CORRINGTON

RCA Victor
Camden 2, N. J.

220.—B. VAN DER POL, "On the non-linear partial differential equation satisfied by the logarithm of the Jacobi theta-functions, with arithmetical applications, I," Nederl. Akad. Wetensch., *Proc.*, s.A., v. 54 [*Indagationes Math.*, v. 13], 1951, p. 261–284.

p. 281 *for* β_{28} = 336 87218 32202 92775 96104 01280
read β_{28} = 436 56892 24858 87663 46104 01280

B. VAN DER POL

12 Chemin Kreig
Geneva, Switzerland

UNPUBLISHED MATHEMATICAL TABLES

151[F].—A. GLODEN, Factorisation of $N^4 + 1$ for isolated values of N between 30000 and 40000, II. Two manuscript pages. Deposited in the UMT FILE.

This constitutes an extension of UMT **144** [*MTAC*, v. 6, 1952, p. 102] and gives 50 new factorisations.

A. GLODEN

11 rue Jean Jaurès
Luxembourg

152[F].—A. GLODEN, *Table of the Least Solution of the Congruence* $2x^2 + 1 \equiv 0 (\mathrm{mod}\ p^2)$ *and Factorisation of the Corresponding Numbers* $2x^2 + 1$. Three manuscript pages. Deposited in the UMT FILE.

The prime p is taken less than 1000.
The largest number $2x^2 + 1$ factored is

$$2(380552)^2 + 1 = 3\cdot 11\cdot 883^2\cdot 11257.$$

A. GLODEN

11 rue Jean Jaurès
Luxembourg

153[F].—A. GLODEN, *Factorisation Table for the Numbers $N^8 + 1$, $N = 500$.* Six typewritten pages. Deposited in the UMT FILE.

The table is an extension of CUNNINGHAM'S[1] table to $N \leq 200$. Of its 500 numbers 147 are completely factored. All unknown factors exceed 600000.

A. GLODEN

[1] A. J. C. CUNNINGHAM, *Binomial Factorisations.* V. 6, London 1923, p. 140–141.

154[F].—F. GRUENBERGER, *Lists of Primes.* Two sheets tabulated from punched cards. Deposited in the UMT FILE.

The list of primes is extended from 50039981 to 50060033. There are 1131 primes between these limits. This is a continuation of a list given in UMT **148** [*MTAC*, v. 6, p. 167].

F. GRUENBERGER

Univ. of Wisconsin
Madison, Wis.

155[F].—R. J. PORTER, *Tables of Irregular Negative Determinants of exponent 3n.* Typewritten manuscript on deposit in the UMT FILE.

The table gives the values of $D < 50000$ for which there is a determinant $-D$ which is irregular with an exponent of irregularity which is divisible by 3. [See DICKSON'S *History*[1] for definition of these terms.]

The table is arranged by thousands. There are 11, 17, 21, $\cdots$, 43 D's in the first, second, $\cdots$, 50th thousand, a total of 1718 D's altogether. Most of these have exponent 3. Only $D = -17561$ has an exponent 6. Thirteen however have exponent 9. These are $-D = 3299$, 6075, 11907, 17739, 23571, 24300, 27675, 29403, 33075, 35235, 41067, 46899, and 47628. All other D's have exponent 3.

The list was constructed by making extracts from some hundreds of the writer's series of determinants of class-number $3k$. To each determinant in these series belongs a class which has the property of duplicating into its own opposite; e.g., the determinant 21481 has a class (149, 71, 178) which duplicates into (26522, 8117, 2485) and thence by reduction to (2485, $-$ 662, 185), (185, $-$ 78, 149) and (149, $-$ 71, 178).

These extracts are filed in numerical order with their corresponding A values (e.g., 149 in the above) and any determinants which have more than one entry of A values against them are irregular (exp. $3n$).

It is found, in practice, that to make extracts from the series for each block of 10,000 determinants takes approximately 40 hours' work.

R. J. PORTER

266 Pickering Road
Hull, England

[1] L. E. DICKSON, *History of the Theory of Numbers,* v. 3, Washington 1927, New York, 1934, Chap. 5.

AUTOMATIC COMPUTING MACHINERY

Edited by the Staff of the Machine Development Laboratory of the National Bureau of Standards. Correspondence regarding the Section should be directed to Dr. E. W. CANNON, 415 South Building, National Bureau of Standards, Washington 25, D. C.

TECHNICAL DEVELOPMENTS

THE USAF-FAIRCHILD SPECIALIZED DIGITAL COMPUTER

The USAF-Fairchild Computer (or "SPEC," for Special Purpose Electronic Computer) was designed and constructed at the NEPA Project[1] in Oak Ridge, Tennessee, to obtain solutions for a specific type of problems. These problems consisted of large systems of linear, algebraic equations involving up to 300 variables.

The SPEC, shown in the frontispiece, is an electronic, fixed-decimal machine, with all numbers lying between − 1 and + 1, and has four-digit capacity, although it will be explained later that eight digits are carried in the accumulator. By specializing the purpose of the computer, construction time, development of special components and circuitry were minimized. It was also possible to simplify the problem insertion so that little skill is required of the operating staff.

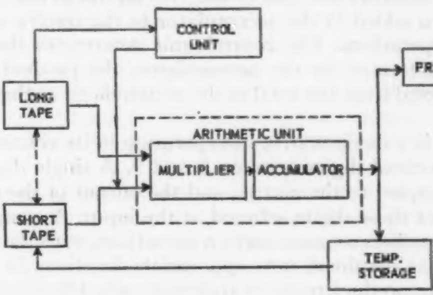


FIG. 1.

The NEPA Computer employs the Gauss-Seidel iteration method for the solution of simultaneous, linear equations. By this method, a sequence of approximate solutions is obtained by successive iterations such that the limits of these solutions are the correct solutions of the equations.

When the problems have been arranged for solution, each of the equations in the set is solved for one of the variables as a function of a constant term and the remaining variables. In this manner, if the order of the set is N, there will be N equations of the following type:

$$X_i = \sum_{j=1}^{j=N} A_{ij}X_j + C_i \text{ where } A_{ii} = 0. \tag{1}$$

The method requires the summing of products and the addition of a constant term C_i. Values are assumed for each variable and substituted successively in the equations where at the conclusion of the evaluation of the right-hand member of each equation the new value for X_i replaces the old.

In the SPEC, two synchronous, magnetic tapes provide the required storage. These are shown in the block diagram of Fig. 1. A short, endless

tape stores the unknowns. As this tape revolves, the unknowns are read in order. Values for the unknowns, altered by successive iterations, will be repeated during succeeding tape revolutions. All the coefficients and constants are stored on the long tape, with accompanying instructions to the control unit. An instruction indicates to the computer which multiplier is to be selected for the number being read from the long tape and also signals the conclusion of each equation, at which time the new result is printed.

A common sprocket drive provides simultaneous reading of the two tapes. The synchronized drive permits arrangement of the coefficients on the long tape and the unknowns on the short tape in such a manner that a coefficient and the unknown by which it is to be multiplied are read by the two inputs simultaneously when they are needed during the course of the computation.

The computer is, basically, a summing device for a series of products. As shown in Fig. 1, the arithmetic unit consists of a multiplier coupled to an accumulator. The numbers received at the two inputs to the multiplier are multiplied and then added in the accumulator to the results obtained from previous similar operations. The control unit determines the sign of the operation to be performed by the accumulator, the product being either added to or subtracted from the total in the accumulator as the computation requires.

The multiplier is a diode matrix incorporating in its connections all the products of two decimal digits between 0 and 9. A single digit from each input number is coupled to the matrix, and the output of the matrix represents the product of those digits selected at the input. A complete product of two four-digit numbers requires sixteen operations, with the partial products being added, as obtained, into appropriate locations in the accumulator. The complete product requires approximately 400 microseconds.

The accumulator capacity is eight decimal digits. The eight-decimal-digit capacity prevents the formation of an excessive error due to repeated roundoffs in the summation of a series of products. Roundoff of the result is accomplished only at the conclusion of a series, prior to the transfer of a result out of the accumulator.

Each decimal digit is represented by a coded group of four binary digits. All operations in the accumulator are binary operations, using the "excess-three" code. The accumulator consists of 32 parallel-fed binary adder circuits. Sufficient inputs are available to handle the number and the required "excess-three" correction factor simultaneously.

At the conclusion of each series of products, and upon command from the control unit, the resulting total is rounded off and transferred both to a printer circuit, which provides a typed copy of the results for the operator, and to the temporary-storage register. This transfer occurs at the end of a sequence of operations, and the short tape is not then in the proper position for recording the new X over the old X. The temporary-storage register holds this X value until the point on the short tape is reached at which this value is to be recorded. The recording will normally occur during the solution of the next equation, and the computer will both record and use the X in the temporary-storage register during the solution of that equation. The recording location on the short tape is determined by the set of instructions placed on the long tape by the operator prior to the solution of the problem.

A total of 2206 vacuum tubes are used in the computer. Over half of these tubes are high-vacuum diodes, types 6AL5, which were used in preference to crystal diodes for matrix circuits.

Design and construction of the SPEC started in July 1949, and the first problem was solved on February 1, 1950. It was in continuous operation at the NEPA Project from the spring of 1950 until February 1951 and is being installed at the Oak Ridge National Laboratory. The computer will be called the "ORACLE" which is derived from Oak Ridge Automatic Computer for Linear Equations.

During the operating period, the computer was in an operating condition for approximately 85% of the total available time, with 15% of the time devoted to testing and servicing. Most machine errors were detected during the solution of problems, although test problems were used for periodic checks to assure proper operation of the machine. Check circuits, with visual and audible indications, are employed to detect improper operation of the machine in the most critical places, but no complete checking system is employed.

Throughout the period of operation, the computer was exceedingly useful as a means of obtaining solutions to systems of simultaneous, linear equations up to the limit of its useful capacity. In addition, problems involving matrix products, Fourier analysis, numerical integration, and matrix inversion were undertaken with considerable savings in time and effort.

The author wishes to acknowledge the outstanding contribution of Mr. V. G. Lewis and his staff of technicians at the NEPA facility in Oak Ridge in the construction and assembly of the computer. Special recognition is due Mr. L. C. Oakes for his valuable contributions in testing, servicing and operating the machine.

J. J. Stone

Oak Ridge National Laboratory
Oak Ridge, Tenn.

[1] Nuclear Energy for the Propulsion of Aircraft Project, conducted by Fairchild Engine and Airplane Corporation under special contract with the United States Air Force.

Discussions

FLOATING OPERATIONS ON THE EDSAC

Summary. The difficulties which arise when programming calculations for large automatic calculating machines which have a fixed decimal point are discussed. This leads to a consideration of the possibility of using floating decimal arithmetic for certain kinds of calculations. A method by which floating decimal arithmetic can be carried out with any fixed decimal-point machine is outlined and the scheme adopted for use with the EDSAC is described in detail.

This scheme is based on a special kind of subroutine which we call *interpretive*. This enables the programmer to use a new order code of his own choice. The 'orders' of a programme drawn up with such a code are selected under the action of the interpretive subroutine and interpreted in terms of sequences of orders which perform the required operations. With the EDSAC

a single address 'order' code has been adopted, similar to the ordinary order code, in which the arithmetical 'orders' are interpreted in terms of floating decimal arithmetic. Special 'orders' for simplifying counting operations and the modification of other 'orders' are provided. The 'order' code is described in detail and an example is given of a programme drawn up using this code. Other topics discussed are the use of auxiliary subroutines with the interpretive subroutine, a method of facilitating programme assembly, and techniques for using the input tape as a form of auxiliary store. Finally the times of operation of the individual 'orders' of the code are given, together with an estimate of the factor by which the calculation time is increased as a result of using floating decimal arithmetic for an entire calculation.

Introduction. One of the most tedious tasks arising in the programming of calculations for solution by fixed decimal-point machinery is to arrange the calculation so that all quantities concerned remain within the limits of the machine and yet are expressed to an accuracy sufficient to ensure the desired precision in the results. Our experience with the EDSAC has shown that if this task can, in some way, be relegated to the machine then the time taken to prepare a programme after a problem has been understood is considerably reduced.

The problem does not arise with machines designed to operate directly with numbers expressed in the floating radix form. Numbers in this form are represented by $a \cdot r^p$. The first machine of this kind was the Bell Telephone Laboratories Relay Computer Model V[1]. This is a decimal machine (that is, $r = 10$) in which $1 > |a| > 0.1$, $19 \geq p \geq -19$ and a is expressed to an accuracy of seven significant figures. Since this was completed all important relay machines have been equipped with similar facilities. No electronic machine of this kind has yet been built but we would remark that in our opinion an electronic machine provided with a floating point arithmetical unit would be a powerful computing instrument even if it had a relatively slow store, a magnetic drum, for example. It would be particularly suitable for a laboratory which had to solve quickly a wide variety of problems as they are presented.

For a fixed decimal-point machine the usual arrangement is to associate scale factors with some or all of the quantities occurring in the calculation. These scale factors are of two kinds. First certain quantities can be associated with *fixed scale factors* which remain unaltered throughout the calculation. Their value must be chosen so that the quantities concerned do not exceed capacity and yet can be represented at all times with sufficient accuracy. This is the problem of providing 'elbow room.' It may not be possible to satisfy both requirements by the use of a single fixed scale factor and it is then necessary to introduce an *adjustable scale factor*, that is, a scale factor which is altered during the course of the calculation. It may be adjusted either continually or occasionally in accordance with some criterion or it may take a preassigned set of values. All these things the coder has to think about if the calculation is to be arranged to use the least possible machine time. The difficulties arising will vary from one calculation to another and may be trivial, moderately complex, or really hard. The individual nature of calculations and the necessity for a mathematical understanding of them, however, makes a general solution to the problem on these lines unattractive to attempt. The only way in which the difficulties may be avoided is to associate every number occurring in the calculation with its own ad-

justable scale factor. The scale factors can be stored most economically if they are powers of 2 or of 10. In the latter case the above scheme virtually amounts to programming a floating decimal point. This is the solution that we feel should be adopted for many kinds of calculation.

The floating decimal form of representation of numbers is especially useful when it is required to evaluate algebraic expressions. For example, to code the evaluation $(au^2 + bu + c)/(du + e)$ with a fixed decimal-point machine can be very troublesome if it is required to maintain accuracy over the entire range of numbers. If the reader has any doubts about this he is advised to try it. If floating decimal arithmetic is used, then the coding requires little thought and is quite direct.

It is the purpose of this paper to describe a method by which floating decimal arithmetic can be carried out on fixed decimal-point machines, and in particular to describe the scheme adopted for the EDSAC.

General considerations. The following remarks are fairly general and apply to almost any machine although certain features mentioned, for example, short and long locations, are made with the EDSAC in mind.

The most convenient way to programme operations on numbers expressed in the floating decimal form is by means of a special type of subroutine which we call ***interpretive***. This enables words similar to those representing ordinary machine orders to be interpreted in terms of floating decimal operations. Such words will be referred to as 'orders' (with quotation marks). Each 'order' resembles an ordinary machine order but is never obeyed as such by the control circuits of the machine. Instead, 'orders' are selected in a definite sequence by the action of the interpretive subroutine and interpreted in accordance with a preassigned 'order' code, by means of sequences of orders which form part of the interpretive subroutine. The selective action of the interpretive subroutine will be referred to as the 'control' (in quotation marks). The advantage of such interpretive subroutines is that the 'order' code may be chosen to suit the convenience of the programmer. There need be no relation between the form of the 'order' code and the ordinary code of the machine. However, we have become so familiar with the ordinary EDSAC order code that a similar single address 'order' code has been adopted for the interpretive subroutines. Interpretive subroutines have also been placed in the EDSAC library for carrying out arithmetical operations on complex numbers and on double-precision numbers.

When representing floating decimal numbers in a fixed decimal-point machine it is most economical to pack both the numerical part and the exponent into a single storage location. This means, of course, that the numerical part of the number has fewer binary digits to represent it than would normally be available. However, they are all significant whereas in fixed decimal-point working digits are thrown away to provide 'elbow room.' In both cases accuracy may be lost owing to cancellation of leading digits at one end of a number and to rounding-off errors at the other.

Two long and two short storage locations are set aside to form a kind of 'arithmetical unit.' One long location holds the numerical part of a number and one short location holds the exponent. Together they form the ***floating decimal accumulator***. In a similar fashion the other long location and the other short location form the ***floating decimal register***.

An arithmetical 'order' first causes a subroutine to select and unpack the operand and place the numerical part and the exponent into a further pair of storage locations—the 'multiplier register.' In the case of an add 'order' the operand is then added to the number held in the floating decimal accumulator. This is done as follows. Let a_0, a_a, and a_s denote the numerical parts of the operand, augend, and sum respectively, and let p_0, p_a, and p_s similarly denote their exponents. Then we have

$$\begin{aligned} a_s 10^{p_s} &= [a_0 + a_a 10^{-(p_0-p_a)}]10^{p_0} \quad & p_0 \geq p_a \\ &= [a_0 10^{-(p_a-p_0)} + a_a]10^{p_a} \quad & p_0 < p_a. \end{aligned}$$

The *subtract* 'order' works in a similar fashion. A *register* 'order' places the operand in the floating decimal register. Other 'orders' enable the product of the operand with a number held in the floating decimal register to be added to or subtracted from the number held in the floating decimal accumulator. The *transfer* order causes the number held in the floating decimal accumulator to be converted to standard form, packed, and finally transferred to the store; the floating decimal accumulator is then 'cleared' by replacing the number held in it by zero, that is, by the special number 010^{-63}. The *input* 'order' causes the two parts of a number to be read from the input tape, packed, and transferred to the store. The *output* 'order' causes the numerical part and the exponent of the number held in the floating decimal accumulator to be printed on the same line in two adjacent columns on the page of the teleprinter.

The use of two separate storage locations for the floating decimal accumulator allows the range and accuracy of numbers held therein to be greater than those held in a single storage location elsewhere. This enables products to be accumulated without loss of accuracy due to intermediate rounding-off errors.

The scheme adopted for the EDSAC. The interpretive subroutine that has been developed to carry out floating decimal arithmetic with the EDSAC will now be described in more detail. Throughout this and the following section references to specific features or conventions used with the EDSAC have been avoided as far as possible but the following terms are used at least once: *initial orders*, *control combinations*, *preset parameters* (not to be confused with the *current parameter* defined below), *short* and *long* locations and the *E* order. Detailed descriptions of all these features can be found in reference 2.

The following abbreviations will be adopted:

- $F(A)$ denotes the floating decimal number held in the floating decimal accumulator
- $F(R)$ denotes the floating decimal number held in the floating decimal register
- $S(mD)$ denotes the long storage location having address m
- $S(mF)$ denotes the short storage location having address m
- $F(mD)$ denotes the floating decimal number held in $S(mD)$
- $F(mF)$ denotes the floating decimal number held in $S(mF)$
- $C(mD)$ denotes the ordinary number held in $S(mD)$
- $C(mF)$ denotes the ordinary number held in $S(mF)$

Number representation. The method described above of packing the numerical part a and the exponent p of a floating decimal number $a \cdot 10^p$ in a

single storage location has been adopted. With the EDSAC it amounts to representing the floating decimal number by the ordinary (fractional) number $p2^{-6} + a2^{-7}$, where a is expressed with an accuracy of 28 binary digits and lies in the range $(1 \geq |a| \geq 0.1)$ and p is an integer such that $63 \geq p \geq -63$. Zero has the special representation 010^{-63}. When numbers are transferred from the floating decimal accumulator to the store they are automatically transferred to the standard representation.

The 'order' code. Since the 'orders' are similar in form to the ordinary machine orders of the EDSAC they will be presented in the conventional form adopted for the latter. The 'orders' of the code fall into two classes, *arithmetical* and *organizational*. The arithmetical 'orders' can refer either to short or to long storage locations according as the 'order' is terminated with the code letter F or D. This is the usual EDSAC convention. If, however, these 'orders' are terminated by $\pi\Delta$ instead of D or Δ instead of F then a number, known as the *current parameter*, held in a certain storage location, will be added to the address of the order before it is obeyed. Among the organizational 'orders' are two—the P and the F orders—which enable the current parameter to be set to some initial value and subsequently adjusted after each cycle of a repetitive operation. In this way an arithmetical 'order' can refer to different numbers during different cycles of the calculation. These orders are described in more detail below. The scheme is similar in some respects to the way in which orders are modified in the Manchester University Electronic Computer Mk. 2. In this machine a built-in facility enables a number held in one of 8 registers—called the B registers—to be added to the address of an order immediately before that order is obeyed.

The arithmetical 'orders'

$A\ m\ D$ Add $F(mD)$ to $F(A)$

$B\ m\ D$ Subtract $F(mD)$ from $F(A)$

$H\ m\ D$ Replace $F(R)$ by $F(mD)$

$V\ m\ D$ Add the product $F(R) \cdot F(mD)$ to $F(A)$

$N\ m\ D$ Subtract the product $F(R) \cdot F(mD)$ from $F(A)$

$D\ m\ D$ Replace $F(A)$ by $F(A)/F(mD)$

$\phi\ m\ D$ Replace $F(mD)$ by $F(A)$ and $F(A)$ by 0.10^{-63}

$L\ m\ D$ Similar to $\phi\ m\ D$ but in addition the non-standard content of the floating decimal accumulator is printed as negative sign or space; exponent (2 figs.); space; negative sign or space; 8 decimal digit fraction. For example, -98.283742 would be printed as $02 - 98283742$.

$\Delta\ m\ D$ Input a sequence of numbers on the tape terminated by X into the locations mD, $(m-2)D$, etc. Each number is punched in the following way: Characters to represent the exponent; sign; numerical part. In the numerical part the decimal point is understood to be before the first digit punched. Any number of digits may be punched. For example, -98.283742 would be punched as $2 - 98283742$. X is punched before the first number of the sequence and then the sequence is copied on to the data tape in the reverse direction. Numbers read into the machine must have exponents in the range -16 to $+15$, so that it can be represented by a single tape character.

The organizational 'orders'

- *M m F* Conditional order: if $F(A) > 0$ transfer 'control' to the 'order' which stands in $S(mF)$; otherwise replace $F(A)$ by $|F(A)|$ and proceed with the next 'order.'
- *C m F* Transfer control of the machine to the order which stands in $S(mF)$, that is, return to the machine order code.
- *X m F* Transfer 'control' to the X-auxiliary whose entry 'order' stands in $S(mF)$. When the X-auxiliary is completed it will return 'control' to the 'order' immediately following the X 'order.'
- *G m F* Transfer 'control' to the 'order' standing in $S(mF)$.
- *P m F*, *F m F* The function of these 'orders' is described below

The P and F 'orders.' These orders are provided to facilitate the coding of cycles of orders and of cycles within cycles, etc. Each cycle of 'orders' or 'loop' is started by a P 'order' and terminated by an F 'order.' The two are said to *correspond.* This correspondence is similar to that between left and right handed brackets in a lengthy algebraic formula if the direction left to right in the formula corresponds to the direction in which 'control' is normally advanced. Associated with each loop of 'orders' is a parameter whose value is set, initially, by the P 'order' and is subsequently adjusted by the F 'order.' The current parameter (see above) associated with any particular 'order' of a loop is the parameter set by the P 'order' which the 'control' encountered last. It is now possible to give a formal description of the function of these 'orders.'

- *P m F* Record a new parameter—the current parameter of the following 'orders'—and set it to the value $-m2^{-15}$
- *F m F* If the current parameter is $< -m2^{-15}$, increase its value by $m2^{-15}$, return 'control' to the 'order' immediately following the corresponding P 'order'; otherwise proceed with the following 'order' after restoring the current parameters to the value it had before the corresponding P 'order.'

The above description should be sufficient to enable a programmer to use the P and F orders in a programme. We now give a detailed description of the means by which the effects of these 'orders' are achieved.

The number of parameters has been limited to 3. This restriction, although not essential, is reasonable as problems involving loops within loops more than three 'deep' are likely to occur very rarely.

The parameters are stored together with the addresses of the locations of the corresponding P 'orders,' in 3 long storage locations $S(hD)$, $S(h + 2)D$, and $S(h + 4)D$. At any stage in the programme $S(hD)$ contains the current parameter, $S(h + 2)D$ the previous parameter, and $S(h + 4)D$ the previous parameter but one.

A P 'order,' for example, $P\ m\ F$ in $S(nF)$ causes $C(h + 4)D$ to be replaced by $C(h + 2)D$, $C(h + 2)D$ to be replaced by $C(hD)$, and $C(hD)$ to be replaced by the number $-m2^{-15} + n2^{-33}$. When the corresponding F 'order,' $F\ q\ F$ is encountered the following operations take place. The size of the current parameter (the number $-m2^{-15}$ in the short storage location $S(h + 1)F$) is tested. If this number is less than $-q \cdot 2^{-15}$ it is increased by the amount $q \cdot 2^{-15}$ and 'control' is transferred to the 'order' which stands in

$S(n+1)F$, where $n \cdot 2^{-15}$ is the number in the short location $S(hF)$. Otherwise $C(hD)$ is replaced by $C(h+2)D$, $C(h+2)D$ is replaced by $C(h+4)D$ and then 'control' proceeds with the next 'order.'

Example. The action of the 'orders' and the context in which they are meant to be used may be more clearly understood from a study of the following programme for the calculation of a root of an algebraic equation by the Newton-Raphson iterative process.

Location of data: let $f(x) \equiv \sum_{r=0}^{r=n} a_r x^{n-r} = 0$ be the equation to be solved.

It is assumed that the coefficients $a_n, a_{n-1}, \cdots, a_0$ are in $S(100\text{D})$, $S(98\text{D})$, $\cdots$, $S(100-2n)\text{D}$. The initial approximation x_0 stands in $S(6\text{D})$ and all subsequent approximations are placed there. A small quantity δ, used in the convergence criterion, stands in $S(8\text{D})$.

Formula used: the iterative formula

$$x_{n+1} = x_n - f(x_n)/f'(x_n)$$

is used. The iteration is arranged to terminate when $|x_n - x_{n-1}| < \delta$. $f(x)$ is calculated from the recurrence relations

$$q_0 = a_0; \quad q_{r+1} = q_r \cdot x + a_{r+1}; \quad q_n = f(x)$$

and $f'(x)$ is calculated from the similar recurrence relations

$$q_0' = 0; \quad q_{r+1}' = q_r' \cdot x + q_r; \quad q_n' = f'(x).$$

The orders of the programme are listed below. For those readers who are not familiar with the EDSAC conventions it may be stated that the code letter θ provides for a system of numbering relative to the order immediately following a control combination G K. Thus, in the example below, the 'order' M 18 θ refers to the 'order' B 8 D.

	G		K		
0	ϕ		D	'clear' floating decimal accumulator	
1	ϕ	10	D	'clear'; $S(10\text{D})$	
2	H	6	D	place x_n in the floating decimal register	
3	P	$2n+2$	F		
4	ϕ	12	D		cycle $n+1$ times
5	V	10	D	$p_{r+1}' = xp'_r + p_r$	
6	A	12	D		
7	ϕ	10	D		
8	V	12	D	$p_{r+1} = xp_r + a_{r+1}$	
9	A	102	$\pi\Delta$		
10	F	2	F		
11	D	10	D	$f(x_n)/f'(x_n)$ to $S(10\text{D})$	
12	ϕ	10	D		
13	B	10	D	$x_{n+1} = x_n - f(x_n)/f'(x_n)$	
14	A	6	D		
15	ϕ	6	D		
16	A	10	D	form modulus of $(x_{n+1} - x_n)$	
17	M	18	θ		
18	B	8	D	subtract δ	
19	*M*	1	θ	discriminate	
20	ϕ		D	'clear' floating decimal accumulator	
21	A	6	D	print the root x	
22	*L*	6	D		

Auxiliary subroutines. To facilitate the coding of entire calculations in floating decimal arithmetic it is useful to extend the range of action of the interpretive subroutines by the addition of auxiliary subroutines for carrying out common numerical processes. The auxiliary subroutines, henceforth referred to simply as *auxiliaries*, are of two kinds. The first kind consists entirely of ordinary orders and will be referred to as *C auxiliaries*. Auxiliaries of the second kind consist largely of 'orders' and hence themselves use the interpretive subroutine. They will be referred to as *X* auxiliaries.

C auxiliaries. These are called in by a *C* 'order' which transfers control to the entry order of the auxiliary. This then carries out the appropriate calculation by means of machine orders. When this has been completed control is transferred back to a point in the interpretive subroutine. This causes the 'control' to resume the obeying of 'orders' starting at the 'order' immediately following the *C* 'order.' Thus the *C* auxiliary can be considered as being an extension of the 'order' code and used as such.

A typical *C* auxiliary is one which replaces $F(A)$ by its square root. The way in which this is done using ordinary orders is as follows. Let p_a and n_a denote the exponent and numerical part of $F(A)$. Two cases arise: If p_a is even, $p_s = p_a/2$ and $n_s = \sqrt{n_a}$; if p_a is odd, then $p_s = (p_a + 1)/2$ and $n_s = \sqrt{n_a/10}$. In both cases the arithmetical operations can be done most simply by using the ordinary techniques of fixed decimal-point working.

C auxiliaries are almost as fast and as economical in storage space as the corresponding routines for fixed decimal-point working. The extra orders required to handle the exponent are largely offset by the orders that would otherwise be required to cater for a large range of the numerical part.

X auxiliaries. These are called in by the special 'order' *X m F*, where *m* is the location of the entry order of the auxiliary. When the *X* auxiliary has completed its part of the calculation 'control' is returned to the 'order' immediately following the *X* 'order' which called the auxiliary into use. This cannot be done by the same method as is used for the *C* auxiliary because the *X* auxiliary itself uses the interpretive subroutine. Instead the *X* 'order' causes the address of the location of the next 'order' to be stored in a certain location for future reference. This record is called the ***link***. At the end of the auxiliary is a *C* 'order' which directs control to a set of orders within the interpretive subroutine which use the link to return 'control' to the main programme.

X auxiliaries can use other *X* auxiliaries. To enable this to be done the *X* 'order' causes a list of links to be kept. At any point in the programme the link at the head of the list refers to the 'order' to which 'control' must be returned after the current subroutine has been finished with. The number of links in the list measures the depth to which 'control' has passed within the auxiliaries. A reference number which is adjusted every time a link is added to or removed from the list records this depth. This enables the link at the head of the list to be selected by those orders of the interpretive subroutine which are called into use when the *X* auxiliary has been completed.

The Directory. Reference to the auxiliaries can be facilitated, if desired, by using the following scheme. The auxiliaries are first enumerated in the order in which they are to be read from the input tape into the store. Thus auxiliary no. 1 is the first auxiliary read from the tape, auxiliary no. 2 is the

second auxiliary read from the tape, and so on. When the auxiliaries have been ordered in this way the m-th auxiliary can be called in by the 'order' $X\ m\ L$ or $C\ m\ L$ (depending on whether it is an X or a C auxiliary).

This is achieved by means of a table of switching orders—called a *directory*—stored in consecutive storage locations beginning with $S(hF)$. Each entry directs 'control' (or control) to the first 'order' (or order) of one of the auxiliaries. If, for example, the m-th auxiliary is an X auxiliary whose first 'order' stands in $S(nF)$, then the m-th entry in the directory, that is, the entry standing in $S(h + m)F$, is the 'order' $g\ n\ F$. Thus to call in the auxiliary 'control' is first transferred, by the 'order' $X\ h + m\ F$, to the 'order' $G\ n\ F$ which in turn transfers 'control' to the first 'order' of the auxiliary. Similarly, if the m-th auxiliary is a C auxiliary, the m-th entry is the ordinary order $E\ n\ F$, and the auxiliary is called in by the 'order' $C\ h + m\ F$.

The directory is assembled by the initial orders as the auxiliaries are read from the tape into the store. At the same time the address of the locations, $S(hF)$ of the first entry in the table is placed in the preset parameter location L so that $X\ m + h\ F$ and $C\ m + h\ F$ can be punched as $X\ m\ L$ and $C\ m\ L$ respectively.

The advantage of the scheme is that the master routine can be drawn up in the final form once the set of auxiliaries have been ordered. It is not necessary for the coder to keep a record of the locations in the store of individual auxiliaries. In effect, the scheme shifts the burden of 'book-keeping' from the programmer to the machine. Similar principles are used by the assembly subroutines (for programme assembly) which already exist in the EDSAC library of subroutines (see reference 2).

The use of the input tape as a form of auxiliary store. The EDSAC is not provided with an auxiliary store so that for many problems the shortage of storage space is a real difficulty. This difficulty may be partly overcome by using the paper tape input medium as a form of auxiliary store. There are two ways in which this may be done.

The first way, used when the storage requirements grossly exceed those available, is to carry out the calculation in stages, using the input tape to store numbers and 'orders' not required throughout the calculation. This method will be referred to as ***piecewise control*** of the calculation.

In the second method the 'orders' of the master routine are placed on the tape and read into the store one at a time, each being obeyed immediately after it has been read. This method is used when the master routine consists of a lengthy sequence of 'orders' not many of which are repeated. If none of these are repeated the scheme takes no longer than putting the complete master routine into the store and then entering it. The principal advantage is that no storage space need be allocated to the master routine. A further advantage is that the progress of the calculation is apparent from the progress of the tape. This method of control will be referred to as ***input control***. In the Manchester University Computer Group, tapes drawn up on these lines are referred to as ***job-steering tapes***.

Piecewise control. To facilitate this mode of working the Δ 'order' and the initial orders can be used as follows.

The Δ 'order' enables sequences of numbers of any length to be read from the tape into the store when required, overwriting, if necessary, information no longer wanted.

If the initial orders are retained intact during the course of the calculation they can be recalled into use by a *C* 'order'. In this way 'orders' (or orders) can be read from the input tape into the store in the usual way at any time during the calculation. The reading of 'orders' can be halted and control transferred back to the interpretive subroutine by a suitable control combination.

The above scheme together with the use of the Δ 'order' enables quite complicated problems to be tackled.

Input control. In this scheme a special *C* auxiliary—the *input control auxiliary*—is used. This causes 'orders' to be read from the tape and interpreted immediately after they are read. Normally they will be of an arithmetical character or will call in auxiliary subroutines. The *C* auxiliary will continue to read and obey such 'orders' until this mode of 'control' is terminated by a suitable *C* 'order' on the tape itself. This directs control to a point within the interpretive subroutine and in this way causes 'control' to be returned to the 'order' immediately following the *C* 'order' which called the input control auxiliary into use.

Times of operations of the 'orders.' The times of execution of the individual 'orders' are as follows:

'orders'	Times of execution
A,B	90 ms.
V,N	105 ms.
H, C, G, X, M, P, F	50 ms.
D	140 ms.
ϕ	80 ms.

The times of operation of the *L* and Δ 'orders' are largely determined by the speed of the input and output units. The teleprinter can print six characters per second and the tape-reader can read about 25 characters per second. These rates allow for the time taken for binary to decimal conversion and vice versa.

For comparison it should be stated that each ordinary order of the EDSAC takes 1½ ms. with the exception of the multiplication orders, *V* and *N*, which take 6 ms. each.

From a direct comparison it would seem that the floating 'orders,' other than those used for reading and writing, are about 60 times as slow as the machine orders and hence that a programme using the interpretive subroutine would be slower by the same factor. This is not altogether true because in such a programme fewer 'orders' are needed than would otherwise be necessary as there are no scale factors to deal with and the techniques for counting and for the modification of 'orders' have been streamlined. Moreover, the time taken by the *C* auxiliaries is about the same as that taken by the corresponding subroutine in fixed decimal-point working.

These factors vary from problem to problem but our experience has shown that the reduction in speed varies from about 20 to 1 to about 4 to 1. The reduction of the time taken to code a problem has to be experienced to be believed!

R. A. BROOKER
D. J. WHEELER

University Mathematical Laboratory
Cambridge, England

The authors wish to thank Dr. M. V. WILKES for his encouragement and advice in preparing this paper.

[1] F. ALT, "A Bell Telephone Laboratories 'Computing Machine—1 and II,'" *MTAC*, v. 3, 1948, p. 1–13 and 69–84.
[2] M. V. WILKES, D. J. WHEELER, & S. GILL, *The preparation of programmes for an electronic digital computer, with special reference to the* EDSAC *and the use of a library of subroutines.* Addison-Wesley Press, Inc., Cambridge, Mass., 1951.

BIBLIOGRAPHY OF CODING PROCEDURE

The National Bureau of Standards is forming a central library of notes, reports and technical publications concerned with programming and coding for electronic digital computers.

The collection of material has in view the possibility of increasing the efficiency in the use of high speed digital computers. The material will include for example coding manuals, computing routines, supervisory routines and codes for generating other codes. Those interested in research and instruction in coding practices are invited to avail themselves of this material.

Computation laboratories are invited to submit material to this library. A list of such acquisitions together with a short descriptive review will appear in the future issues of *MTAC*; they will be numbered serially for reference. Material should be sent to the attention of J. H. WEGSTEIN, Computation Laboratory, National Bureau of Standards, Washington 25, D. C.

Some material of this kind has already been included or reviewed in *MTAC*. This bibliography begins with references to eleven such items. The last four items are new.

1. S. LUBKIN, "Decimal point location in computing machines," *MTAC*, v. 3, p. 44–50.
2. HERMAN H. GOLDSTINE & JOHN VON NEUMANN, *Planning and Coding of Problems for an Electronic Computing Instrument*. Institute for Advanced Study, Princeton, New Jersey.
   Part II, v. 1, 1947, 69 p., 21.6 × 27.9 cm. [*MTAC*, v. 3, p. 54–56]
   Part II, v. 3, 1948, iii + 23 p., 21.6 × 27.9 cm. [*MTAC*, v. 3, p. 541–542]
   Part II, v. 2, 1948, 68 p., 7 figs., 21.6 × 27.9 cm. [*MTAC*, v. 4, p. 44–46]
3. FLORENCE KOONS & S. LUBKIN, "Conversion of numbers from decimal to binary form in the EDVAC," [*MTAC*, v. 3, p. 427–431]
4. M. V. WILKES, "Programme design for a high-speed automatic calculating machine," *Jn. Sci. Inst. and Phys. in Industry*, v. 26, 1949, p. 217–220. [*MTAC*, v. 4, p. 116]
5. ANON., *Description and Use of the ENIAC Converter Code*. Ballistic Research Laboratories, *Technical Note* no. 141, Aberdeen Proving Ground, Maryland, November 1949, 23 pages, mimeographed. [*MTAC*, v. 4, p. 170–171]
6. R. W. HAMMING, "Error detecting and error correcting codes," *Bell System Technical Jn.*, v. 29, 1950, p. 147–160. [*MTAC*, v. 5, p. 40–41]
7. M. V. WILKES, "Automatic computing," *Nature*, v. 166, December 2, 1950, p. 942–944. [*MTAC*, v. 5, p. 171]
8. S. GILL, "The diagnosis of mistakes in programmes on the EDSAC," R. Soc. London, *Proc.*, v. 206A, 1951, p. 538–554. [*MTAC*, v. 6, p. 49–50]

9. M. V. Wilkes, D. J. Wheeler & S. Gill, *The Preparation of Programs for an Electronic Digital Computer*. Addison-Wesley Press, Inc., Cambridge, Mass., 1951, 167 p., 22.9 × 15.2 cm. Price $6.00 [*MTAC*, v. 6, p. 51]
10. R. A. Brooker, S. Gill & D. J. Wheeler, "The adventures of a blunder," [*MTAC*, v. 6, p. 112–113]
11. David P. Perry, "Minimum access programming," [*MTAC*, v. 6, p. 172–182]
12. University of Illinois Research Board, *University of Illinois Electronic Digital Computer Library Routines*. This volume is an expandable library of the routines which have been prepared for the Illinois machine and at present contains perhaps fifty routines. It contains among many other such routines as the Wheeler Leapfrog Test, the solution of simultaneous linear equations, solution of a system of first order differential equations, post mortem, integration routine, address search routine, eigenvalues and eigenvectors of a symmetric matrix, and square root.
13. Eckert-Mauchly Computer Corporation, *Programming Manual for the UNIVAC System*. This describes the UNIVAC and how to code for it. The manual includes chapters on: flow charts, an illustrative problem (Denominated Payroll routines), floating decimal and double precision routines, elementary mathematical problems, input and output operations, editing, problem preparation.
14. Remington-Rand Corporation, *Roster of UNIVAC Problems*.
15. J. D. Goodell, "The foundations of computing machinery," *Jn. of Computing Systems*, v. 1, 1952, p. 1–13.

J. T.

Bibliography Z

1000. Martin Gardner, "Logic machines", *Scientific American*, v. 186, March 1952, p. 68–73.

This is an historical account of the development of these machines beginning with the period of George Boole, the English scientist who first conceived the idea of symbolic logic as a system designed for the more efficient handling of problems. The paper discusses one of the devices of Lull, a 13th century Spanish mystic, who probably was the first to construct a machine to handle logical problems. The device consisted of a set of wheels having a common center. Around the edge of the disks were letters representing various ideas. When the wheels were rotated, various combinations of ideas would appear for further investigation. This primitive machine prompted Charles Stanhope in the 18th century to invent the Stanhope demonstrator for solving syllogisms. Later William Stanley Jevons developed his "abecedarium," a method of applying Boole's approach. In the case of a syllogism all possible combinations of the three terms *A*, *B* and *C* are listed (the negative of these terms being represented by *a*, *b*, and *c*). In solving a problem all classes inconsistent with the premises are crossed out—leaving only the true combinations. These are then examined to determine the relationship between the terms in an effort to weed out the false conclusions. Jevons went on to construct his more rapid "logical piano" on the same principle.

Since 1900 many devices for solving syllogisms have been invented. With the publication of an article by CLAUDE E. SHANNON of M.I.T. entitled, "A symbolic analysis of relay and switching circuits," it was proved that circuits could be constructed in series and in parallel making use of the binary system to indicate the two truth values which correspond to basic Boolean relations. The Kalin-Burkhart Calculator, a small low-cost device, is the first in history to use electrical circuits for solving problems in symbolic logic. It is similar in principle to the Jevons piano but excells in that it can handle a greater number of problems far more intricate in complexity than those solved on Jevons' machine.

At present these machines have limited application; probably their chief value lies in the fields of business procedures and of electronic computers. In the latter field, especially, decisions involving considerations of logic are often necessary in setting up problems to be solved on the machine. It has even been suggested that with built-in probability values the machine could after considering all possibilities make important logical decisions. The question of whether or not such a machine could develop a creative imagination has been the subject of much fantastic speculation.

EDITH T. NORRIS

NBSMDL

1001. JOHN D. GOODELL, "The foundations of computing machinery," *The Journal of Computing Systems*, v. 1, June 1952, p. 1–13. TENNY LODE, "The realization of a universal decision element," *The Journal of Computing Systems*, v. 1, June 1952, p. 14–22.

These articles are, according to their authors, the beginning of a series discussing the logical design of computing machinery in terms of Boolean algebra or, more generally, symbolic logic. They treat mainly the two-valued functions of one and of two two-valued variables, and particularly the recognition and realization of complete subsets of the two-valued functions. Functions of many arguments, variable functions and many-valued logics are very briefly discussed.

R. D. ELBOURN

NBSEC

1002. HARVARD UNIVERSITY, COMPUTATION LABORATORY, *Annals*, v. 27: *Synthesis of Electronic Computing and Control Circuits*. Cambridge, Mass., Harvard University Press, 1951, 278 p., 20 × 27 cm. Price $8.00.

The bewildering complexity of electronic computing and control circuits and the profusion of means available for achieving the same functional performance make it extremely desirable to have 1) a concise and unique notation for writing the function of a circuit so that functionally identical requirements or circuits can readily be recognized, 2) a simple translation of this unique notation into a non-unique one corresponding more closely to the actual circuits so that one has a facile shorthand for manipulating a functional requirement into various possible realizations and 3) a method for finding a most economical realization of a required function.

It is well known that Boolean algebra is useful for 1) and 2), but this book is the most extensive treatment of these applications which has appeared.

The method of "minimizing charts" introduced in chapter 5 is a significant contribution to the very formidable problem of most economical synthesis. At the present state of the theory, minimizing charts and tables included in the book provide a rather complete solution for circuits with one output and four or five input variables. For six or seven variables the charts become quite unwieldy. Some useful methods are given for circuits with multiple outputs and circuits containing signal delay or storage devices, but there remains the necessity for much arbitrary decision in their synthesis, and the economy problem is wide open.

The second half of the book covers systems for binary coding decimal digits and the design of a variety of adders and multipliers mostly of the coded decimal type.

The Boolean algebra of this book uses only the logical product $X\ Y$ and negation, written $X' = 1 - X$, because the arithmetic of these operations coincides with ordinary arithmetic. While this is a complete system, the reviewer would have included the logical sum, in the belief that the difficulty of remembering $1 + 1 = 1$ is outweighed by having symmetrical notations for both types of rectifier circuits and by having the powerful duality principle to manipulate expressions; however this is merely personal preference.

Although the efficient designing of electronic computing and control circuits has only begun to be systematized, anyone who wishes to use the best known techniques and possibly to contribute some new ones must certainly consult this volume.

R. D. ELBOURN

NBSEC

1003. OFFICE OF NAVAL RESEARCH, *Digital Computer Newsletter*, v. 4, July 1952, 9 p.

The contents are as follows:

1. Whirlwind I
2. The SEAC
3. Naval Proving Ground Calculators
4. Moore School Automatic Computer (MSAC)
5. The Logistics Computer
6. Aberdeen Proving Ground Computers
   - The ENIAC
   - The EDVAC
   - The ORDVAC
7. The Circle Computer
8. Electronic Computer Corporation (ELECOM—100)
9. The CADAC (CRD 102)
10. The ACE Pilot Model, England
11. The Ferranti Computer at Manchester University, England

Data Processing and Conversion Equipment

1. Digital to Analogue Converters
2. Oscillograph Trace Reader
3. Flying Typewriter

1004. INSTITUTE OF RADIO ENGINEERS, "Standards on Electronic Computers: Definitions of Terms, 1950." Institute of Radio Engineers, *Proc.*, v. 39, no. 3, 1951, p. 271–277.

This is the first report of the IRE Subcommittee on Definitions of Electronic Computer Terms. R. SERRELL was chairman of this subcommittee, J. W. FORRESTER headed the Electronic Computers Committee, and J. G. BRAINERD was chairman of the Standards Committee. There is a total of approximately 135 terms for which definitions are given. About 30% of these terms are arithmetical or logical, over one-half relate specifically to digital devices, and approximately 20 terms concern analog devices. (It might be of interest to note that a new IRE subcommittee is at present reviewing these definitions.)

H. D. HUSKEY

Wayne University
Detroit, Michigan

1005. IBM Scientific Computation Forum, *Proc.* edited by H. R. J. GROSCH, 1948, 126 pages. 22 × 28.5 cm.

This volume refers to one of a series of meetings arranged from time to time by the IBM Corp. (cf. the following reviews). The contents are as follows:

1. "Evaluation of higher order differences on the Type 602 Calculating Punch" by FRANK M. VERZUH
2. "Differencing on the Type 405 Accounting Machine" by GERTRUDE BLANCH
3. "The use of optimum interval mathematical tables" by H. R. J .GROSCH
4. "Punched card techniques for the solution of similtaneous equations and other matrix operations" by WILLIAM D. BELL
5. "Two numerical methods of integration using predetermined factors" by LELAND W. SPRINKLE
6. "Integration of second order linear differential equations on the Type 602 Calculating Punch" by N. ARNE LINDBERGER
7. "Integration of the differential equation $\frac{d^2P}{dr^2} = P \cdot F(r)$ using the Type 601 Multiplying Punch" by PAUL HERGET
8. "Some elementary machine problems in the sampling work of the census" by A. ROSS ECKLER
9. "IBM applications in industrial statistics" by CUTHBERT C. HURD
10. "Some engineering applications of IBM equipment at the General Electric Company" by FRANK J. MAGINNISS
11. "Planning engineering calculations for IBM equipment" by BEN FERBER
12. "A survey of the IBM project at Beech Aircraft Corporation" by JOHN KINTAS
13. "Aerodynamic lattice calculations using punched cards" by HANS KRAFT
14. "Dynamics of elliptical galaxies" by JACK BELZER, GEORGE GAMOW and GEOFFREY KELLER
15. "Application of punched cards in physical chemistry" by GILBERT W. KING

16. "Application of punched card methods to the computation of thermodynamic properties of gases from spectra" by LYDIA G. SAVEDOFF, Jack Belzer and H. L. JOHNSTON
17. "Calculation of the equilibrium composition of systems of many constituents" by STUART R. BRINKLEY, JR. and ROBERT W. SMITH, JR.
18. "Punched card calculating and printing methods in the Nautical Almanac Office" by FREDERICK H. HOLLANDER
19. "Programming and using the Type 603–405 Combination Machine in the solution of differential equations" by GEORGE S. FENN
20. "Applications of punched card equipment at the Naval Proving Ground" by CLINTON C. BRAMBLE
21. "Use of the IBM relay calculators for technical calculations at Aberdeen Proving Ground" by JOSEPH H. LEVIN
22. "Simultaneous linear equations" by FRANCIS J. MURRAY
23. "Computation of shock wave refraction on the selective sequence electronic calculator" by HARRY POLACHEK
24. "Computation of statistical fields for atoms and ions" by L. H. THOMAS

At the meeting reported here, most of the papers described details of the use of IBM machines for various problems in numerical mathematics, which are adequately characterized by their titles. A few of the papers present surveys of the computing work done in certain organizations on IBM machines. The paper of Fenn introduces the 603–405 combination (cf. the following review). Murray's paper refers to his special-purpose machine for solving linear equations. Grosch's article is of more special interest to table-makers.

F. L. ALT

NBSCL

1006. IBM Seminar on Scientific Computation, *Proc.* edited by CUTHBERT C. HURD, November 1949, 109 pages. 22 × 28.5 cm.

The contents of this volume are as follows:

1. "The dynamics of nuclear fission" by DAVID L. HILL
2. "Monte Carlo calculations" by WILLIAM W. WOODBURY
3. "Modification of the Monte Carlo method" by HERMAN KAHN
4. "Analyzing exponential decay curves" by ALSTON S. HOUSEHOLDER
5. "On the distribution of Kolmogorov's statistic for finite sample size" by Z. W. BIRNBAUM
6. "The IBM Card-Programmed Electronic Calculator" by CUTHBERT C. HURD
7. "Stochastic methods in quantum mechanics" by GILBERT W. KING
8. "Calculation of resonance energies" by GEORGE E. KIMBALL
9. "Cam design calculations on the Card-Programmed Electronic Calculator" by E. A. BARBER
10. "Calculation of the equilibrium composition of homogeneous multicomponent systems" by STUART R. BRINKLEY, JR. and ROBERT W. SMITH, JR.
11. "Eigenvalue problems related to the Laplace operator" by DONALD A. FLANDERS and GEORGE H. SHORTLEY

12. "Numerical solution of partial differential equations of parabolic type" by L. H. THOMAS
13. "Solutions of the wave equation" by PAUL HERGET
14. "Sampling methods applied to differential and difference equations" by JOHN H. CURTISS

The conference about which this volume reports was on a higher scientific level than other similar conferences arranged by the IBM Company and requires more detailed review. Not all the papers in this volume are primarily of interest to the numerical analyst. For example, Kahn's paper is an exposition of Monte Carlo methods as applied to definite integrals and integral equations, with emphasis on importance sampling, but without reference to machine methods or to any of the deeper questions of numerical analysis. The main purpose of Birnbaum's paper is "to point out that in view of the development of high-speed sequence computing equipment, the tabulation of the exact probability distribution of Kolmogorov's statistic for finite N has become practically feasible, and to propose that such a tabulation should be carried out." The article by G. W. King is principally concerned with the formulation of certain quantum mechanical problems in terms of finite-difference equations, and thence in terms of random walks; some particular numerical results are given as an indication of the rate of convergence of the Monte Carlo method when used for the solution of such problems. Similarly, Kimball's paper is principally interesting for the mathematical formulation of a physical problem rather than for the method of solution.

Other papers contain more substantial contributions to our understanding of machine methods of computation. The article by Hill outlines the theory of the liquid-drop model of nuclear fission and gives details of the numerical techniques used on the SSEC for certain prototype computations in this field. The paper by Woodbury describes shielding computations by a Monte Carlo method, run on the "405–603 Combination," the famous machine improvised by Woodbury and Toben by combining two standard IBM machines. This combination was the predecessor of the IBM Card-Programmed Calculator. The approximation of a given function by a linear combination of exponentials is the subject of Householder's paper. Several methods—graphical, least squares, iterations based on Poisson-distributed errors—are compared. C. C. Hurd gives a functional description of the Card-Programmed Calculator. This is the so-called "Model I" which was in process of construction at the time of the presentation of this paper. Barber's paper describes the evaluation of certain polynomials. Brinkley and Smith report on progress in their long-standing problem which consists mathematically of the solution of numerous systems of nonlinear algebraic equations. They describe a special method applicable in certain degenerate cases and mention briefly the use of the Card-Programmed Calculator. The paper of Flanders and Shortley is a discussion of several iterative methods for the computation of wave functions; particularly methods employing matrix operators which are obtainable by forming polynomials of simpler operators. Herget gives a sketchy description of the numerical integration of a system of second order ordinary differential equations on an IBM 602-A calculator.

The most important papers in this volume, from the standpoint of numerical analysis, are those of L. H. Thomas and J. H. Curtiss. The former is an

exhaustive treatment of questions of approximating parabolic differential equations by finite-difference methods, the errors in the solution caused by such approximation, stability and rate of convergence. While the discussion is purposely nonrigorous, it is fundamental and complete. An unpublished but frequently quoted method due to J. VON NEUMANN is used.

The paper by J. H. Curtiss, which occupies almost one-fourth of the volume, is the only one written with the care and rigor customary in mathematical publications, while the rest of the volume is in the nature of an exchange of information on techniques, with emphasis on timeliness, effectiveness in getting results, and "know-how" rather than on scientific requirements. The paper begins with a historical survey of the field, which is shown to be much older than its currently fashionable name of "Monte Carlo Method." There follows an exposition which brings together results, some of which had previously been widely scattered and relatively inaccessible in the literature on mathematical physics and theory of probabilities. Significant new results are presented on the standard error of the solution, the number of samples (random walks) necessary to achieve a given accuracy, the mean length of a walk and thus the expected amount of computing effort necessary for a given problem, and on importance sampling.

The editorial work is excellent throughout the volume, except for the discussions at the end of the papers, where misprints like the following occur: p. 27, 1.6, for "Prof. Katz and Dr. Dunning" read "[M.] KAC and [M. D.] DONSKER"; p. 78, l. 22, for "Dr. Foster" read "[G. E.] FORSYTHE."

F. L. ALT

NBSCL

1007. IBM Industrial Computation Seminar, edited by CUTHBERT C. HURD, September 1950, 103 p. 22 × 28.5 cm.

Participants in this seminar discussed fundamental computational methods used by research engineers and scientists in a wide variety of research problems. Particular attention was drawn to computational techniques developed in the fields of chemistry and petroleum. The contents of this volume are as follows:

1. "The role of the punched card in scientific computation" by WALLACE J. ECKERT
2. "Machine calculation of the plate-by-plate composition of a multicomponent distillation column" by ASCHER OPLER and ROBERT C. HEITZ
3. "Continuous distillation design calculations with the IBM Card-Programmed Electronic Calculator" by ARTHUR ROSE, THEODORE J. WILLIAMS and WILLIAM S. DYE, III
4. "Application of automatic computing methods to infrared spectroscopy" by GILBERT W. KING
5. "Correlation and regression analysis" by E. L. WELKER
6. "Pile-driving impact" by EDWARD A. SMITH
7. "Punched card mathematical tables on standard IBM equipment" by ELEANOR KRAWITZ
8. "The solution of simultaneous linear equations using the IBM Card-Programmed Electronic Calculator" by JUSTUS CHANCELLOR, JOHN W. SHELDON and G. LISTON TATUM

9. "Two applications of the IBM Card-Programmed Electronic Calculator: The Gauss-Seidel method of solution of simultaneous linear equations; Approximating the roots of a polynomial equation" by I. C. LIGGETT
10. "Matrix by vector multiplication on the IBM Type 602-A Calculating Punch" by ELEANOR KRAWITZ
11. "Numerical solution of two simultaneous second-order differential equations" by WALTER H. JOHNSON
12. "Numerical evaluation of integrals of the form $\int_a^b f(x)g(x)dx$" by JOHN W. SHELDON
13. "The use of orthogonal polynomials in curve fitting and regression analysis" by JACK SHERMAN
14. "General-purpose ten-digit arithmetic on the IBM Card-Programmed Electronic Calculator" by STUART R. BRINKLEY, JR., G. L. WAGNER and R. W. SMITH, JR.
15. "Remarks on distillation calculations" by JOHN W. DONNELL
16. Some applications of the Monte Carlo method:
  "Matrix inversion on the IBM Accounting Machine" by Ascher Opler
  "Remarks on finding roots of, and inverting, a matrix" by GILBERT W. KING
  "Remarks on the Monte Carlo method" by CUTHBERT C. HURD
17. "Plotting punched card data using the IBM Type 405 Accounting Machine" by PAUL T. NIMS
18. "A method for evaulating determinants and inverting matrices with arbitrary polynomial elements by IBM punched card methods" by L. E. GROSH, JR. and E. UDSIN

The article by Chancellor, Sheldon and Tatum describes a method for solving simultaneous linear algebraic equations up to and including order twenty-one using the Crout method. With this procedure, a 20 × 20 matrix can be inverted in two hours. The prescribed method assumes a model I CPC equipped with one (941) auxiliary storage. The article by Brinkley, Wagner and Smith, Jr., describes a general purpose ten-digit arithmetic set-up for the model I CPC. In addition to the elementary arithmetic operations, square root and certain combinations, one can automatically select alternative computational routines and also select input data.

R. K. ANDERSON

NBSCL

1008. IBM, Industrial Computation Seminar, *Proc.* edited by Cuthbert C. HURD, August 1951, 148 p. 28.5 × 22 cm.

The participants of this seminar were research engineers and scientists representing computing facilities which employ IBM card-programmed electronic calculators. The contents of this volume are as follows:

1. "Application of the IBM Card-Programmed Electronic Calculator to engineering procedures at the Glenn L. Martin Company" by W. B. KOCH
2. "Reduction of six-component wind tunnel data using the IBM Card-Programmed Electronic Calculator, Model II" by M. L. LESSER
3. "IBM Card-Programmed Electronic Calculator operations using a Type 402–417BB and 604–2" by H. E. TILLITT, M. KENYON and B. OLDFIELD

4. "The Combomat" by J. D. MADDEN
5. "The IBM Type 604 Electronic Calculating Punch as a miniature electronic calculator" by P. T. NIMS
6. "General-purpose floating point control panels for the IBM Card-Programmed Electronic Calculator" by N. A. PATTON, K. BERGER and L. R. TURNER
7. "Catapult takeoff analysis" by J. R. LOWE
8. "Computation of loan amortization schedules on the IBM Card-Programmed Electronic Calculator" by C. H. GUSHEE
9. "Techniques for handling graphical data in the IBM Card-Programmed Electronic Calculator" by W. D. BELL
10. "Calculation of the flow properties in an arbitrary two-dimensional cascade" by J. T. HORNER
11. "Automatic calculation of the roots of complex polynomial equations using the IBM Card-Programmed Electronic Calculator" by J. GALLISHAW, JR.
12. "A recursion relation for computing least square polynomials over moving-arcs" by GEORGE R. TRIMBLE, JR.
13. "Numerical solution of second-order non-linear simultaneous differential equations" by HENRY S. WOLANSKI
14. "Matrix inversion and solution of simultaneous linear algebraic equations with the IBM Type 604 Electronic Calculating Punch" by GEORGE W. PETRIE, III
15. "The determination of eigenvectors and eigenvalues of symmetric matrices" by EVERETT C. YOWELL
16. "An application of the IBM Card-Programmed Electronic Calculator to analysis of airplane maneuvering horizontal tail loads" by LOGAN T. WATERMAN
17. "Fifth-order aberration in an optical system" by RUTH K. ANDERSON
18. "Theory of plastic vibrations of helicopter fuselages" by PETER F. LEONE
19. "Machine procedure for computation of elastic vibrations of helicopter fuselages" by WILLIAM P. HEISING

Nims describes how a 604 Electronic Calculating Punch may be used as a small sequence-controller calculator. The 604 can perform a different arithmetic operation on each card. In addition it is possible for the machine to shift, look up entries in a table of functions, calculate square roots and stop when certain check conditions do not hold.

Patton, Berger and Turner have designed general-purpose floating point control panels for.the CPC which give the coder a great deal of flexibility. In addition to the basic arithmetic operations, one has three independent program fields and four numerical fields (per card), automatic extraction of integral roots from second to seventh, eight kinds of conditional operations and machine stop with automatic list cycles on all improper operations (division by zero, overflow).

Gallishaw's technique for finding complex roots of a polynomial is an iterative procedure employing synthetic division and Newton's method.

Using standard trial values, it takes from twenty to twenty-five minutes to find all the roots of a complex 7th degree equation. If any of the roots are approximately known in advance, they can be used instead of the standard trial values, thus cutting down the solution time. The process uses eight-digit arithmetic throughout.

Petrie has devised a method for matrix inversion and solution of simultaneous linear equations using a 604 Electronic Calculating Punch, a tabulator and two reproducers.

The procedure consists of a continuous flow of cards through these four machines, the cycle being repeated N times to invert an Nth order matrix with approximately N^2 cards each time. The size of the matrix is unlimited, and a check operation is included.

R. K. ANDERSON

NBSCL

1009. IBM *Technical News Letter No. 3*, Applied Science Department, December 1951, 106 pages, 22 × 28 cm.

The contents are as follows:

1. "Mass spectrometer calculations on the IBM Type 602-A Calculating Punch" by W. H. KING, JR. and WILLIAM PRIESTLY, JR.
2. "Computations of inverse matrices by means of IBM machines" by JACK SHERMAN
3. "Linear equations and matrix inversion" by ERIC V. HANKAM
4. "Matrix and vector algebra" by Eric V. Hankam
5. "Notes on the IBM Type 604 Electronic Calculating Punch and Type 602-A Calculating Punch" by Eric V. Hankam
6. "Stop controls for the IBM Type 604 Electronic Calculating Punch," by BRUSE MONCREIFF
7. "An eight-digit general-purpose control panel" by WILLIAM P. HEISING
8. "Interpolation on the IBM Card-Programmed Electronic Calculator" by STUART R. BRINKLEY, JR. and G. L. WAGNER.

The article by King and Priestly describes in detail the control panels used in performing a matrix multiplication of the 20th order and in subsequent normalization steps. The general-purpose board described by Heising (for use with either the IBM Type 604 Electronic Calculator or the IBM Card-Programmed Calculator) calculates, in addition to the basic arithmetic operations, $\sqrt{A}$, $\exp(\pm A)$, $\log(1 \pm A)$, $\sin A$, $\cos A$, $\sinh A$, $\cosh A$, arcsin A, arctan A, arcsinh A, and arctan A.

R. K. ANDERSON

NBSCL

NEWS

Association for Computing Machinery.—The University of Toronto Computation Centre acted as host to the Association for the fall meeting held September 8 and 9, 1952. The University had on display for its guests the three computers located at the Computation Centre, namely, FERUT, a newly installed large-scale computer built by Ferranti Limited which was undergoing tests at the time; UTEC, the model computer built at the University;

and BERTIE, a noughts and crosses machine built by Rogers Majestic Electronic Laboratories. The program for the meeting was as follows:

Sept. 8, 1952 10:00 a.m. to 12 noon

General Session	S. B. WILLIAMS, President, ACM, Chairman
Welcoming address	S. SMITH, President, Univ. of Toronto
Pure and Applied programming	M. V. WILKES, Director, University Mathematical Laboratory, Cambridge, England
MANIAC	N. METROPOLIS, E. F. KLEIN, W. ORVEDAHL, J. R. RICHARDSON, H. B. DEMUTH, J. B. JACKSON, Los Alamos Scientific Laboratory
Chebyshev polynomials in the solution of large scale linear systems	C. LANCZOS, NBSINA

Sept. 8, 1952, 2:00 p.m. to 5:00 p.m.

Programming	C. C. HURD, IBM, New York, Chairman
Interpretative sub-routines	J. M. BENNETT, D. G. PRINZ, M. L. WOODS, Ferranti, Ltd., Manchester, England
Machine aids to coding	E. J. ISAAC, NRL
Computer aids to code checking	I. C. DIEHM, NBSCL
Input scaling and output scaling for a binary calculator with decimal output	E. F. CODD, H. L. HERRICK, IBM
Compiling routines	R. K. RIDGWAY, Eckert-Mauchly Division of Remington RAND, Inc., Philadelphia
Function table techniques in programming	R. F. SHAW, Electronic Computer Corp., New York

Sept. 8, 1952, 2:00 p.m. to 5:00 p.m.

Computing machines	K. F. TUPPER, Dean, Faculty of Applied Science and Engineering, Univ. of Toronto, Chairman
The logical design of the Oak Ridge digital computer	C. L. PERRY, Oak Ridge National Laboratory
The Oak Ridge automatic computer	J. C. CHU, Argonne National Laboratory
Designing a low cost general-purpose computer	W. E. DOBBINS, Computer Research Corp., Hawthorne, California
The University of Toronto model electronic computer	R. F. JOHNSTON, Computation Centre, Univ. of Toronto
A description of the electronic computer at the Institute for Advanced Study	G. ESTRIN, Institute for Advanced Study, Princeton

Banquet, Sept. 8, 1952, 7:30 p.m.

Guest Speaker	
The St. Lawrence Seaway Project	R. H. SAUNDERS, Chairman, Hydro-Electric Power Commission of Ontario

Sept. 9, 1952, 9:00 a.m. to 12:00 noon

Machine calculations	MINA REES, ONR, Chairman
Errors in iterative solutions of linear systems	A. S. HOUSEHOLDER, Oak Ridge National Laboratory

The numerical solution of a partial differential equation on the IBM Type 701 Electronic Data Processing Machines	D. W. LADD, J. W. SHELDON, Applied Science Dept., IBM, New York
A numerical solution of the helium wave equation with the SEAC	J. H. WEGSTEIN, NBSCL
Matrix inversion by partitioning	M. LOTKIN, R. REMAGE, BRL, Aberdeen Proving Ground
The use of subroutines on SEAC for numerical integrations of differential equations and for Gaussian quadrature	P. RABINOWITZ, NBSCL
Use of continued fractions in high-speed computing	D. TEICHROEW, NBSINA

Sept. 9, 1952, 9:00 a.m. to 12:00 noon

Storage devices	J. A. RAJCHMAN, RCA, Chairman
The testing of cathode ray tubes for use in the Williams type storage system	A. ROBINSON, Ferranti, Ltd., Manchester England
Williams tubes selection program	J. C. CHU, R. J. KLEIN, Argonne National Laboratory
Improvement of Williams memory reliability	R. SCHUMANN, Argonne National Laboratory
Ferro electric materials as storage elements for digital computers and switching systems	J. R. ANDERSON, BTL, New York
An experimental rapid access memory using diodes and capacitors	A. W. HOLT, NBSCML

Sept. 9, 1952, 2:00 p.m. to 5:00 p.m.

Logical design	W. H. WATSON, Head Dept. of Physics, Univ. of Toronto, Chairman
Symbolic synthesis of digital computers	I. S. REED, MIT
Logical or non-mathematical programmes	C. S. STRACHEY, National Research Development Corp., London, England
A simplified universal Turing machine	E. F. MOORE, BTL, New Jersey
Simple learning by a digital computer	A. G. OETTINGER, The Computation Laboratory, Harvard Univ.
An analysis by arithmetical methods of a calculating network with feedback	L. C. ROBBINS, Burroughs Adding Machine Company, Philadelphia

Sept. 9, 1952, 2:00 p.m. to 5:00 p.m.

Engineering	B. V. BOWDEN, Ferranti, Ltd., Manchester, England
A high-speed magnetic-core output printer	B. M. GORDON, R. N. NICOLA, Laboratory for Electronics, Inc., Boston
Development of computer components and systems	W. S. ELLIOTT, Research Laboratories of Elliott Brothers (London), Ltd., Borehamwood, England
An electronic analogue machine for computing the roots of algebraic equations of degrees through the eighth	L. LOFGREN, Research Institute of National Defense, Radio Department, Stockholm, Sweden
Analogue calculation of polynomials and their zeros	M. G. SCHERBERG, J. F. RIORDAN, Wright Air Development Center, Wright-Patterson Air Force Base, Ohio
Operating efficiencies and characteristics of the computing machines at Aberdeen Proving Ground	H. SPENCE, Ballistic Research Laboratories, Aberdeen Proving Ground
Installation of a large electronic computer	L. R. JOHNSON, Hdqs., U. S. Air Force

NBSNAML.—On May 15, 1952, at Washington, D. C., the National Applied Mathematics Laboratories sponsored a meeting of Mathematical Tables in the Light of Electronic Computers. The program for the meeting was as follows:

Work in Progress	J. Todd, NBSCL, Chairman
E. T. Goodwin, Royal Society Mathematical Tables Committee	
J. P. Wong, RAND Corporation	
M. Abramowitz, NBSCL	
G. Blanch, NBSINA	

(There was also a written communication from M. Picone, Instituto Nazionale per le Applicazione del Calcolo, Rome.)

Current Needs	F. L. Alt, NBSCL, Chairman
Physics	G. Breit, Sloane Physics Dept., Yale Univ.
Aerodynamics	C. W. Jones, Dept. of Math., Liverpool Univ.
Military applications	L. S. Dederick, Ballistic Research Laboratories, Aberdeen Proving Ground
Statistics	C. Eisenhart and D. Teichroew, NBSSEL
Is the preparation of a single volume 7-place table of $J_n(x)$ of integral order up to 120, at interval .01, desirable?	R. C. Archibald, Brown Univ.
Are tables of Bessel functions of the second kind and of orders higher than 20 desirable?	R. C. Archibald, Brown Univ.

Long Term Policy	P. M. Morse, MIT, Chairman

Round table discussion of which the principal speakers were:

C. C. Hurd, IBM Corporation
D. H. Lehmer, NBSINA
E. T. Goodwin, Royal Society Mathematical Tables committee
J. H. Curtiss, NBSNAML

Editorial Matters	D. H. Lehmer, NBSINA, Chairman
Index of Statistical Tables	D. L. Wallace, Dept. of Mathematics, Princeton Univ.
Index of Mathematical Tables	C. W. Jones, Dept of Mathematics, Liverpool, University
Editing of Mathematical Tables	G. M. Clemence, Naval Obsrevatory, Washington, D. C. and D. H. Sadler, His Majesty's Nautical Almanac Office
The preparation of a bibliography of calculating machines developed from lists of Aiken, Travis and an Australian physicist, from Patent Office records, and from other sources.	R. C. Archibald, Brown Univ.
An organization to assemble for MTAC tabular material news from the very numerous research centers of this and other countries	R. C. Archibald, Brown Univ.

OTHER AIDS TO COMPUTATION

ANALOGUE CALCULATION OF POLYNOMIAL AND TRIGONOMETRIC EXPANSIONS

Introduction: The use of polynomials and trigonometric expansions in science and engineering is well known, and hence the importance of machine application for their rapid evaluation does not need strong defense. In fact, the multiplicity of special purpose polynomial computers that have been designed and built points to the great desire for machine treatment of these functions. Although the method described in this paper is applicable to digital as well as analogue machines, the original intent of the authors was slanted towards the use of the analogue machine, and hence the presentation has an analogue machine background. The significance and importance of the new method is its simplicity and its use of standard computing machine equipment.

The Vector Multiplier: The key process which leads to the new method is the automatic multiplication of an arbitrary pair of vectors. The present paper will be limited to the treatment of the complex vector $(a + ib)$. The product of two such vectors, $z_1 = x_1 + iy_1$ and $z_2 = x_2 + iy_2$, may be written

$$z_3 = z_1 z_2 = (x_1 x_2 - y_1 y_2) + i(x_1 y_2 + x_2 y_1), \tag{1}$$

and the operation performed by machine as shown in Figure 1.

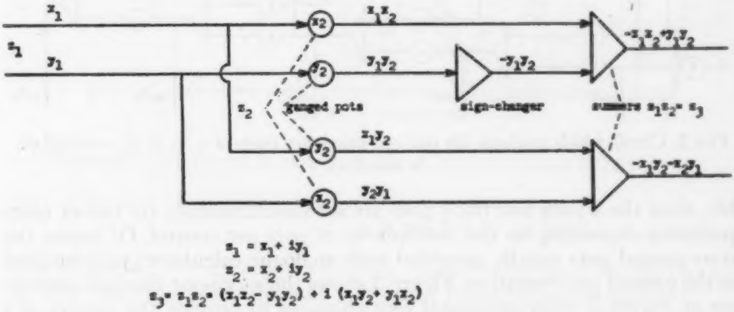


FIG. 1. The Vector Multiplier.

In terms of operation on either the Reeves REAC or the comparable Goodyear Electronic Machine, x_1 and y_1 in the figure represent input voltages to a pair of ganged potentiometers or a set of four simple potentiometers (hereinafter referred to as pots) set respectively at x_2 and y_2. The outputs of these pots are, except for sign, the four terms on the right of equation (1). The term y_1y_2 is put through a sign changer, and then the terms are put into a pair of summers to obtain outputs which are the negatives of the real and imaginary parts of z_3. The amplifiers used on the machines referred to automatically change the sign of their summed inputs and hence

we find the complex vector multiplier described above giving an output vector which is the negative of the product z_3. This is in no way an impediment in the operation since many operations will call for sign changes, and hence, on the average the same sign change equipment would be needed even if the vector product operation produced the product without sign change. In what follows we shall call the input z_1 the multiplicand and the pot setting z_2 the multiplier.

Polynomials in the complex variable z with real coefficients: It is now fairly obvious how successive application of the vector multiplication will produce the successive powers $z^2, z^3, z^4, \cdots, z^m$. Figure 2 shows a part of the circuit diagram for a fifth degree polynomial. Unit voltages enter at the left and the successive output voltages representing respectively z, $-z^2$, z^3, $-z^4$ and z^5 are shown by the vertical output lines. If ganged pots are avail-

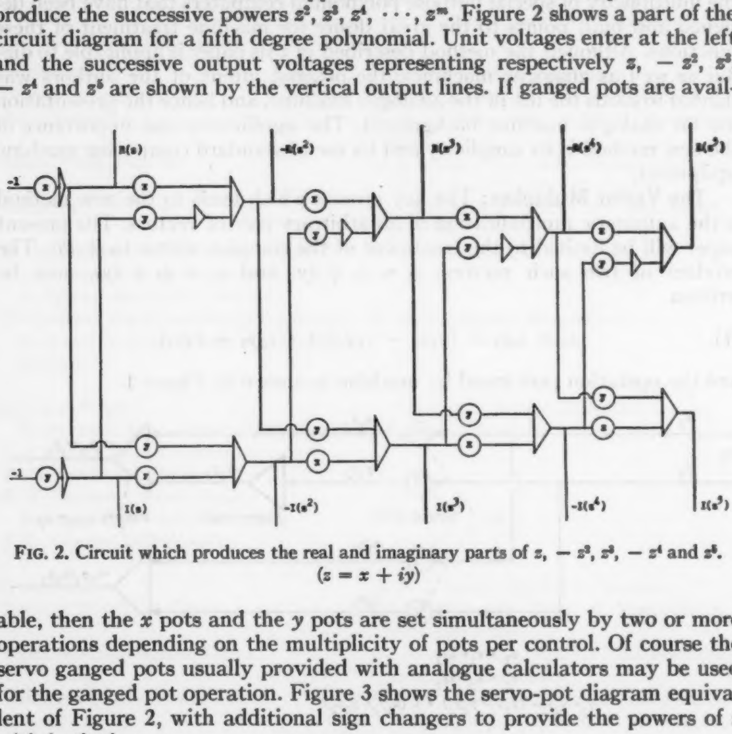


FIG. 2. Circuit which produces the real and imaginary parts of z, $-z^2$, z^3, $-z^4$ and z^5. $(z = x + iy)$

able, then the x pots and the y pots are set simultaneously by two or more operations depending on the multiplicity of pots per control. Of course the servo ganged pots usually provided with analogue calculators may be used for the ganged pot operation. Figure 3 shows the servo-pot diagram equivalent of Figure 2, with additional sign changers to provide the powers of z with both signs.

The outputs $R(z)$, $-R(z^2)$, $R(z^3)$, $-R(z^4)$ and $R(z^5)$, i.e., the real parts of the successive powers indicated, are respectively put through a set of pots which multiply them by the polynomial coefficients $a_1, a_2, a_3, \cdots, a_5$. The outputs of the coefficient pots, including a pot representing the polynomial constant, with appropriate signs, then go into a summing amplifier whose output defines the real part of the polynomial or its negative depending upon the computing requirements. A comparable process is used to calculate the imaginary part of the polynomial.

Polynomials in z with complex coefficients: In this instance it is only a matter of making vector multiplications of the coefficients a_m into the

corresponding vector terms $R(z^m) + iI(z^m)$ in place of using simple pot multiplication. The summing amplifiers used in the latter multiplications may of course be used for all or part of such multiplications depending upon the amplifiers input facilities. The real and imaginary parts of the polynomial may be projected on the screen of the cathode ray tube so that a visual display of the calculated points becomes available. Through this display one discovers rather quickly characteristics of the polynomial under study such as dominant terms and zeros.

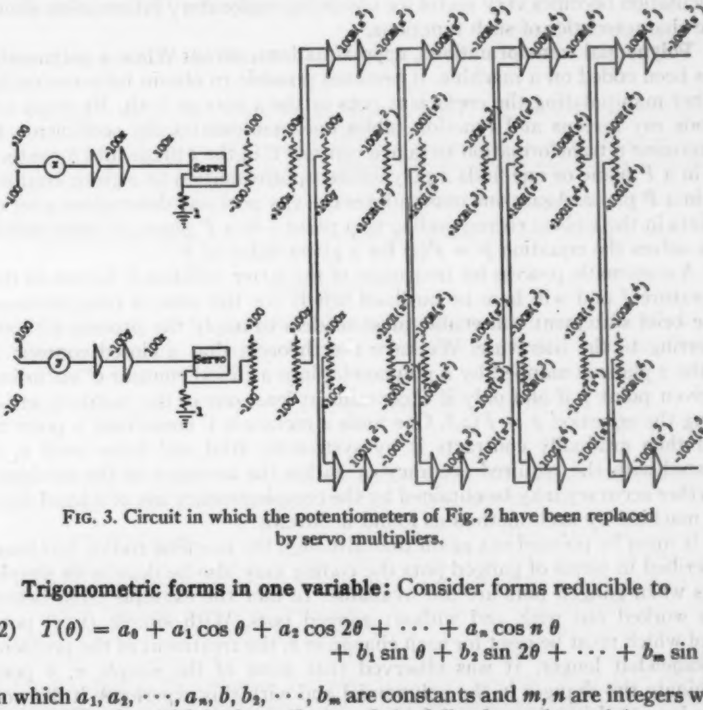


FIG. 3. Circuit in which the potentiometers of Fig. 2 have been replaced by servo multipliers.

Trigonometric forms in one variable: Consider forms reducible to

$$(2)\quad T(\theta) = a_0 + a_1 \cos\theta + a_2 \cos 2\theta + \cdots + a_n \cos n\theta + b_1 \sin\theta + b_2 \sin 2\theta + \cdots + b_m \sin m\theta$$

in which $a_1, a_2, \cdots, a_n, b, b_2, \cdots, b_m$ are constants and m, n are integers with $m \geq n$. The form (2) is the real part of the following polynomial:

$$(3)\quad a_0 + (a_1 - ib_1)z + (a_2 - ib_2)z^2 + \cdots + (a_r - ib_r)z^r + \cdots + (a_m - ib_m)z^m,$$

in which $a_{n+1}, a_{n+2}, \cdots, a_m$ are all zero. To evaluate the trigonometric form for any specific range of θ one has but to evaluate the polynomial (3) over an x, y range corresponding to $\cos\theta$, $\sin\theta$.

The resolver servos available on the machines referred to would facilitate such treatment of the independent variables. In any case, the use of the sine and cosine columns in the trig table would make it relatively simple to range the variable appropriately. The treatment of mixed expansions of terms of the type $r^n \cos n\theta$, $r^n \sin n\theta$ is obvious.

Polynomials in Several Variables: We consider first a polynomial $P(z_1, z_2)$ in two variables (z_1, z_2). In this case the successive powers in z_1 and z_2 are respectively formed by the method described above and the required cross products such as z_1^n, z_2^n are formed by the required vector multiplication. The vector cross product terms are then multiplied by the appropriate scalar or vector coefficient and the results fed into summing amplifiers to form the real and imaginary parts of $P(z_1, z_2)$. Treatment of the more general case is obvious. In the case of polymonials in several variables this machine calculation becomes very useful for obtaining exploratory information about the characteristics of such functions.

Polynomial transformations, approximations, zeros: When a polynomial has been coded on a machine, it becomes possible to obtain information by either manipulating the coefficient pots or the z pots or both. By using cathode ray screens and function tables one manipulates the coefficients to determine a transformation to take a contour C in the z plane into a contour C' in a P plane or one finds a polynomial approximation to a given contour C' in a P plane. Again one manipulates the x, y pots and determines a set of points in the z plane corresponding to a point p in a P plane, in other words one solves the equation $p = P(z)$ for a given value of p.

A systematic process for treatment of the latter problem is known in the literature[1, 2] and will here be outlined briefly for the sake of completeness. The brief statement will enable most readers to apply the process without referring to the literature. We have the theorem that a closed contour C in the z plane is mapped by a polynomial into a closed contour C' enclosing a given point p if and only if C contains at least one of the points z_r satisfying the equation $p = P(z_r)$. One finds a rectangle C containing a point z_r and then gradually contracts it by systematic trial and error until z_r is located with the required accuracy or within the accuracy of the machine. Further accuracy may be obtained by the complementary use of a hand digital machine by such method as found in MILNE.[3]

It must be pointed out again that although the machine coding has been described in terms of ganged pots the coding may also be done with simple pots when ganged pots are not available. In fact the example cited below was worked out with and without ganged pots. With simple (x, y) pots all of which must be reset for each change in z, the treatment of the problem is somewhat longer. It was observed that some of the simple x, y pots dominate the changes in the polynomial and with this experience it did not take long to drive the polynomial values in the required direction.

Example: The polynomial equation

$$P(z) = z^5 - z^4 - 3z^3 + z^2 + 12z + 5 = 0$$

was solved on both Reeves and Goodyear Analogue Machines. One root was located in the rectangle whose vertices were $(0, 0)$, $(2, 0)$, $(2, 2i)$ and $(0, 2i)$. The rectangle was systematically contracted until the root was located with an optimum of accuracy. The polynomial was then reflected in the origin and a pair of roots located in the same rectangle. Since one of these was clearly a real root it was located with an optimum of accuracy by an obvious systematic search on the real segment of the rectangle. The first contraction of this rectangle which left out the axis of reals gave a con-

tour in the P plane which simply inclosed the origin. It was then a matter of further contractions to obtain the best estimate of the root. The table below gives a comparison of the roots as calculated by both analogue and digital methods.

Analogue Method	$-.454$	$1.94 \pm 960i$	$-1.21 \pm .954i$
Digital Method	$-.4517$	$1.935 \pm .9466i$	$-1.216 \pm .9595i$

MAX G. SCHERBERG
JOHN F. RIORDAN

Wright-Patterson Air Force Base
Ohio

This paper was written at the Flight Research Laboratory, Wright Air Development Center, Wright-Patterson Air Force Base, Ohio.

[1] L. BAUER & S. FIFER, "The solution of polynomial equations on the REAC," *Symposium I on REAC Techniques*. U. S. Navy Special Devices Center and Reeves Instrument Corporation, New York, 1951, p. 31–36.

[2] E. C. TITCHMARSH, *Theory of Functions*. Oxford University Press, 1939, Second Edition, p. 116.

[3] W. E. MILNE, *Numerical Calculus*. Princeton University Press, 1949, p. 53–57.

BIBLIOGRAPHY Z

1010. J. EULER & R. LUDWIG, "Zwei Nomogramme zum Gebrauch bei Messungen mit optischen Pyrometern," *Zeitschrift f. angew Physik*, v. 2, 1950, p. 362–373.

The nomograms involve two-dimensional plotting for the inputs.

F. J. M.

1011. H. GLUBRECHT, "Elektrisches Rechengerät für Gleichungen höheren Grades," *Zeitschrift f. angew. Physik*, v. 2, 1950, p. 1–8.

The apparatus is set up to represent both the z and w plane and the roots of the polynomial are located by exploration. The null point of the polynomial is obtained by means of a special cathode ray tube arranged so that when both the real and imaginary parts of w are zero an electron beam goes through a narrow hole and signals the presence of a zero. As far as practical, quantities are represented by voltages having complex significance, thus multiplication is by means of a mixing tube and the resolution into real and imaginary parts is accomplished by a passive network associated with the ordinary type of cathode ray tube upon which z and w are represented. Normally a circle is mapped in the z plane and its w image also is given. The amplitude of this circle can be either varied by hand or swept automatically. The article discusses also two other apparatuses of this type due to TISCHNER and RASCH.

F. J. M.

1012. E. H. GAMBLE & B. W. HATTEN, "Design and analysis of a conservative dynamic load simulator," *Jn. Appl. Phys.*, v. 22, 1951, p. 1250–1257.

The design of a load simulator for the dynamic testing of actual control-system components such as aircraft autopilots is described. The use of an electrical computer for determining the load and a hydraulic servomecha-

nism for actually applying the load to the test unit results in greater flexibility than a load simulator in which the loading is determined mechanically. A feature of this simulator which should prove of great importance in the testing of large servomechanisms is the fact that virtually all the energy absorbed from the unit being tested is returned to the electrical lines and need not be dissipated as heat.

RICHARD C. BOOTON, JR.

Massachusetts Institute of Technology
Cambridge 39, Massachusetts

1013. G. A. KORN & T. M. KORN, *Electronic analog computers* (D-c *analog computers*). McGraw-Hill Book Co., New York, 1952, xv + 378 p., 15 × 23 cm. Price $7.00

This book is concerned with d.c. analog equipment of the type manufactured by Reeves, Goodyear and Boeing and to a certain extent with that of Philbrick. There is also a clear intention to indicate how less expensive versions of these machines may be set up for special purposes. Two special installations, the Curtiss Wright Analog Computer and the RAND analog facility, are discussed.

The first three chapters discuss the principles of these devices, practical set up procedure and a variety of specific applications. The fourth chapter is concerned with the theory of linear computing networks and specifically the necessity for using amplifiers and the effect of this use on the solution. The fifth chapter discusses the design of d.c. amplifiers for computer applications. Various simple types of such amplifiers are discussed as well as the modern drift stabilized amplifiers as developed by WILLIAMS, TARPLEY & CLARK, and the GOLDBERG amplifier. The titles of Chapters 6, 7 and 8 are, respectively, "Multipliction and Function Generation," "Auxiliary Circuits and Computer Operation," and "The Design of Complete D-c Analog-computer Installations."

The case for the use of these computers is skillfully presented and the authors' concern with the accuracy of individual components is admirable. However, the book discusses only briefly the possibilities for large network computers and presumably many other types of analogs were deliberately excluded from the scope, for instance, a.c. analogs, continuous analogs such as electrolytic tanks and conducting sheets, and the many types of stress and strain analogs.

F. J. M.

1014. M. TRIBUS, "Intermittent heating for aircraft ice protection with application to propellers and jet engines," ASME *Trans.*, v. 73, 1951, p. 1117–1130.

The author uses an electrical analog computer for the study of deicing of airplane parts. Thermally speaking, the problem is that of an idealized fin in transient heat flow. Heat is generated in the body which is to be deiced, for example the propeller, by means of electric heating wires; the heat is lost to the surrounding environment across a boundary conductance. The paper deals in considerable detail with the problems of deicing but mentions only

very briefly the computer used. No reference is made to the large amount of literature on the paper.

As far as computing technique goes the main point of interest is an electronic circuit used to represent the boundary conductance which is not constant but rather a function of temperature. This nonlinear condition has heretofore on other computers been represented in discrete finite steps. The author shows in figure 8 a circuit for representation of such boundary conductance. The check of the computations with actual experiments is rather unsatisfactory possibly because of poor assumption of physical constants of the system; the correction of assumptions determines in analog computers the validity of the result. The author, in designing the computing circuit, disregards a number of influences (for example the thermal resistance of the ice).

VICTOR PASCHKIS

Heat and Mass Flow Analyzer Laboratory
School of Engineering
Columbia University, New York, N. Y.

1015. F. WEINER, "Further remarks on intermittent heating for aircraft ice protection," ASME *Trans.*, v. 73, 1951, p. 1131–1137.

The paper deals with the deicing of a propeller of ovoid cross section. The butt end of this ovoid will be called A, the extension, B. Heat is generated by electric heaters only in the region A and is conducted to the region B. A cross section through the propeller in the region A shows (proceeding inwards) the steel blade (extending all the way to B), a layer of nylon, the heater, another layer of nylon, and finally sponge rubber. Disregarding the heat flow along the propeller length, the problem is two dimensional.

The greatest problem in solving two dimensional problems on analog computers, one on which little information is available, is that of how to section the body in which heat flow occurs. The authors disregard the thermal resistance across the thickness of the steel, and along the length of the nylon and heater layers. The sponge rubber is represented by a number of parallel sections, ending in a fictitious center with zero volume (Fig. 3 of the paper). This design is not discussed or analyzed; thus, the most crucial problem, from a computational view-point, is not dealt with in the paper. Regarding deicing, the paper shows the desirability of using high rates of energy production, which results in a lower total heat consumption than heating at a lower rate.

VICTOR PASCHKIS

Heat and Mass Flow Analyzer Laboratory
School of Engineering
Columbia University, New York, N. Y.

NOTES

143.—ANALYTICAL APPROXIMATIONS. [EDITORIAL NOTE: In Note 139, *MTAC*, v. 6, p. 251–253, CECIL HASTINGS has described the RAND *Collection of Illustrative Approximations*. The interest that these approximations have aroused during the past year is considerable. It is hoped to publish as Notes

from time to time additional examples of such approximations contributed by our readers. To encourage this hope, Mr. Hastings is submitting a dozen new examples, prepared with the assistance of Mr. James P. Wong and Mrs. David K. Hayward. These differ from the Rand Collection in form, especially since they do not give illustrative error curves. For convenience in future references we are numbering these approximations consecutively.]

(1) Square Root: To better than 1 part in 12 over $.1 \leq x \leq 10$,
$$x^{\frac{1}{2}} \doteq (1 + 4x)/(4 + x).$$

(2) Pearson Cosine Transformation: To .003 over $0 \leq x \leq 1$,
$r(x) = \cos(\pi/(1 + x^{\frac{1}{2}})) \doteq (-1 - 4x + 5x^2)/(1 + 8x + 6x^2)$.
$r(x^{-1}) = -r(x)$ can be used to obtain function values over $1 \leq x \leq \infty$.

(3) Common Logarithmic Function: To better than .005 over $.1 \leq x \leq 1$, $\log x \doteq -.076 + .281\,x - .238/(x + .15)$.
This approximation is the result of a request for a very simple formula to use in the reduction of certain data.

(4) Common Logarithmic Function: To better than .000,004 over $1 \leq x \leq 10$,
$\log x \doteq \frac{1}{2} + .86857y + .29059y^3 + .15783y^5 + .20269y^7$,
where $y = (x - \sqrt{10})/(x + \sqrt{10})$.

(5) Common Logarithmic Function: To better than .000,000,015 over $1 \leq x \leq 10$,
$\log x \doteq \frac{1}{2} + .8685888y + .2895497y^3 + .1731159y^5 + .1314381y^7 + .0547562y^9 + .1832415y^{11}$,
where $y = (x - \sqrt{10})/(x + \sqrt{10})$.

(6) Inverse Tangent: To better than .005 over $-1 \leq x \leq 1$,
$$\arctan x \doteq x/(1 + .28x^2).$$

This approximation is the result of a request for a very simple formula to use in the reduction of certain data.

(7) Descending Exponential Function: To better than .000,000,11 over $0 \leq x \leq \infty$,
$e^{-x} \doteq (1 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5)^{-8}$,
where $a_1 = .125{,}000{,}204$, $a_2 = .007{,}811{,}604$, $a_3 = .000{,}326{,}627$, $a_4 = .000{,}009{,}652$ and $a_5 = .000{,}000{,}351$.

(8) Incomplete Gamma Function Type Integral: To better than .000,000,1 over $0 \leq x \leq 1$,
$$F(x) = \int_1^\infty e^{-t}t^{-x-1}dt$$
$$\doteq [.219384+x(.024717+.000803x)]/[1+x(.558651+.090584x)].$$
$F(x) + (x + 1)F(x + 1) = e^{-1}$ can be used to obtain function values outside the range indicated.

(9) Exponential Integral of Negative Argument: To better than .000,000,1 over $10 \leq x \leq \infty$,
$$xe^x \int_x^\infty t^{-1}e^{-t}dt$$
$$\doteq [1.15198 + x(4.03640 + x)]/[4.19160 + x(5.03637 + x)].$$

(10) Segmental Area Function: To better than .0012 over $-1 \leq x \leq 1$,

$$A(x) = \int_{-x}^{x} (1 - t^2)^{\frac{1}{2}}\, dt \doteq 2.0083x - .4160x^3 + .1604x^5 - .1808x^7.$$

In terms of elementary functions, $A(x) = \arcsin x + x(1 - x^2)^{\frac{1}{2}}$.

(11) Segmental Area Function: To better than .00016 over $-1 \leq x \leq 1$,

$$A(x) = \int_{-x}^{x} (1 - t^2)^{\frac{1}{2}}\, dt$$
$$\doteq (1.99916x - 2.39484x^3 + .58673x^5)/(1 - 1.03472x^2 + .15634x^4).$$

(12) Segmental Area Function: To better than .000,016 over $-1 \leq x \leq 1$,

$$A(x) = \int_{-x}^{x} (1 - t^2)^{\frac{1}{2}}\, dt$$
$$\doteq x(1.999872 + 4.143151\eta - 3.153670\eta^2 - 1.430807\eta^3)/$$
$$(1 + 2.901498\eta - 1.811287\eta^2 - 1.098016\eta^3),$$

where $\eta = x^2/(5 - 4x^2)$.

CECIL HASTINGS, JR.

RAND Corporation
1700 Main Street
Santa Monica, California

144.—ZEROS OF THE DERIVATIVE OF BESSEL FUNCTIONS OF FRACTIONAL ORDER. The NBS Computation Laboratory[1] has published extensive tables of Bessel functions of fractional order, $J_\nu(x)$, $\pm \nu = \frac{3}{4}, \frac{2}{3}, \frac{1}{3}, \frac{1}{4}$, and zeros of $J_\nu(x)$ have been tabulated by ABRAMOWITZ[2] and by the Computation Laboratory when the latter was known as the Mathematical Tables Project.[3] (The zeros in the last two sources are also given in the first, p. 384–385.) This present note gives the first seven or eight positive zeros of the first derivative of those Bessel functions. Following standard notation, $j'_{\nu,s}$ will be used to denote the s-th positive root of $J_\nu'(x) = 0$. The zeros $j'_{\nu,s}$ were obtained from the published values[1] of $J_\nu(x)$, and this table includes all such zeros within the range of tabulation of $J_\nu(x)$ itself (i.e., not exceeding 25).

These zeros were computed to the maximum accuracy obtainable from the NBS tables. All entries are given here to seven decimal places; although the seventh decimal place is not absolutely guaranteed, it has a high probability of being correct. The zeros were computed using the 5-point case of two different formulas for inverse interpolation for the derivative, given in SALZER[4] on p. 214 and p. 215. They were also checked by (a) calculating an approximate expression for the error in $j'_{\nu,s}$, (b) recomputing the last $j'_{\nu,s}$ given here by the asymptotic formulas for each ν (see below), and (c) computing the seventh divided difference of $j'_{\nu,s}$ as a function of ν, using a formula in Salzer.[5] (This divided difference check was not fully applicable to the first few zeros.)

For zeros > 25, the following asymptotic formulas, whose coefficients were calculated from the general expression for $j'_{\nu,s}$ in WATSON,[6] will give at

least seven decimal accuracy:

$$\pm\nu = \tfrac{3}{4},\quad j'_{\nu,s} = y - .65625000y^{-1} - .54931641y^{-3}$$
$$\text{where } y = \pi s - \begin{cases} .39269908 & \nu = -\tfrac{3}{4} \\ 1.17809724 & \nu = \tfrac{3}{4} \end{cases}$$

$$\pm\nu = \tfrac{2}{3},\quad j'_{\nu,s} = y - .59722222y^{-1} - .41380530y^{-3}$$
$$\text{where } y = \pi s - \begin{cases} .26179939 & \nu = -\tfrac{2}{3} \\ 1.30899694 & \nu = \tfrac{2}{3} \end{cases}$$

$$\pm\nu = \tfrac{1}{3},\quad j'_{\nu,s} = y - .43055556y^{-1} - .07507073y^{-3}$$
$$\text{where } y = \pi s + \begin{cases} .26179939 & \nu = -\tfrac{1}{3} \\ (-1.83259571) & \nu = \tfrac{1}{3} \end{cases}$$

$$\pm\nu = \tfrac{1}{4},\quad j'_{\nu,s} = y - .40625000y^{-1} - .03108724y^{-3}$$
$$\text{where } y = \pi s + \begin{cases} .39269908 & \nu = -\tfrac{1}{4} \\ (-1.96349540) & \nu = \tfrac{1}{4} \end{cases}$$

To obtain $j'_{\nu,s}$ for any other ν that is less than one in absolute value, the present table may be used in conjunction with a special table of interpolation coefficients,[1] p. 393–413. The user is cautioned that for interpolation as well as for forming divided differences, the values of $j'_{\nu,s}$ for $\nu < 0$ are not continued into $j'_{\nu,s}$ for $\nu > 0$, but into $j_{\nu,s+1}$.

Mrs. RUTH E. CAPUANO and Miss MARY M. DUNLAP assisted in the computations.

HERBERT E. SALZER

NBSCL

Table of $j'_{\nu,s}$

s	$\nu = -\frac{3}{4}$	$\nu = -\frac{2}{3}$	$\nu = -\frac{1}{3}$	$\nu = -\frac{1}{4}$	s
1	2.47861 49	2.65267 49	3.27468 22	3.41838 81	1
2	5.77630 68	5.92026 00	6.47892 00	6.61491 38	2
3	8.95866 64	9.09725 75	9.64204 42	9.77606 19	3
4	12.11945 77	12.25581 13	12.79457 06	12.92770 62	4
5	15.27226 12	15.40738 64	15.94278 34	16.07542 28	5
6	18.42121 33	18.55556 21	19.08881 57	19.22113 82	6
7	21.56801 08	21.70182 42	22.23359 29	22.36569 56	7
8	24.71348 00	24.84690 21			8

s	$\nu = \frac{3}{4}$	$\nu = \frac{2}{3}$	$\nu = \frac{1}{3}$	$\nu = \frac{1}{4}$	s
1	1.51433 70	1.40121 80	0.90999 85	0.76906 15	1
2	4.97223 54	4.85063 49	4.35291 38	4.22515 79	2
3	8.16610 90	8.04140 90	7.53529 41	7.40675 25	3
4	11.33027 35	11.20403 00	10.69360 09	10.56453 27	4
5	14.48452 05	14.35735 04	13.84430 89	13.71489 82	5
6	17.63422 27	17.50643 40	16.99164 33	16.86199 56	6
7	20.78145 96	20.65322 86	20.13718 52	20.00736 48	7
8	23.92720 84	23.79864 53	23.28166 09	23.15170 93	8

[1] NBSCL, ***Tables of Bessel Functions of Fractional Order.*** V. 1, New York, Columbia University Press, 1948.
[2] M. Abramowitz, "Zeros of certain Bessel functions of fractional order," *MTAC*, v. 1, p. 353–354.
[3] Mathematical Tables Project, National Bureau of Standards, "More zeros of certain Bessel functions of fractional order," *MTAC*, v. 2, p. 118–119.
[4] H. E. Salzer, "Formulas for finding the argument for which a function has a given derivative," *MTAC*, v. 5, p. 213–215.
[5] H. E. Salzer, "The checking of functions tabulated at certain fractional points," *MTAC*, v. 2, p. 318–319.
[6] G. N. Watson, *A Treatise on the Theory of Bessel Functions.* Cambridge University Press, 1944, p. 507.

145.—An Example in the Use of the Differential Analyser. In a recent article Sprague[1] has discussed, as an example, a differential analyser setup for solving the equation

$$w'' - ww' - wt = 0,$$

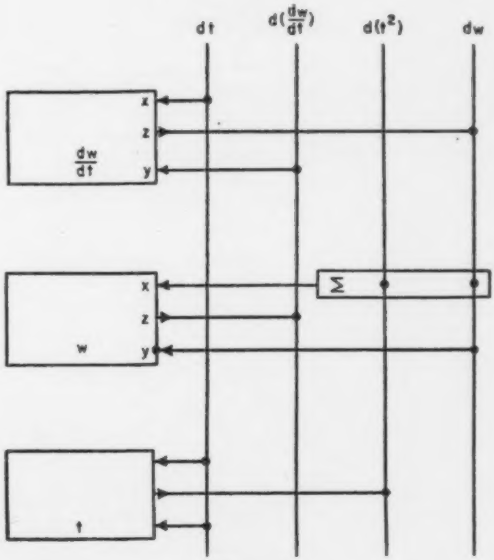


Figure 1

primes denoting differentiation with respect to t. There are several simpler ways of setting up this equation. One is to take the once-integrated form

$$w' = \int_0^t wd(w + \tfrac{1}{2}t^2) + w'(0)$$

and so place it on the analyser in accordance with Figure 1. This uses three integrators in lieu of the six shown in Figure 3 of Sprague's paper.

J. G. L. Michel

National Physical Laboratory
Teddington, Middlesex
England

[EDITORIAL NOTE: Mr. SPRAGUE informs us that an additional integrator will be required in case the differential analyzer is of digital type, thus the above method would require four integrators.]

[1] R. E. SPRAGUE, "Fundamental concepts of the digital differential analyzer method of computation," *MTAC*, v. 6, p. 41–49.

146.—TWO NEW MERSENNE PRIMES. The program described in Notes **131**(c) and **138** [*MTAC*, v. 6, p. 61, 204] has been continued. Two more Mersenne primes, $2^{2203} - 1$ and $2^{2281} - 1$, were discovered by the SWAC on October 7 and 9, 1952. The time required for either of the tests is one hour. This makes 17 Mersenne primes, and a corresponding number of perfect numbers, now known. They are $2^n - 1$ for $n = 2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203$, and 2281.

D. H. L.

CORRIGENDA

v. 6, p. 129, l. 22 *for* $r_i(1 - \rho_{i-1})$ *read* $r_i(1 + \rho_{i+1})$
v. 6, p. 132, l. − 6 and − 18 *for* THOMPSON *read* THOMSON
v. 6, p. 152, l. −5 *for* 8 *read* 9
v. 6, p. 187, l. 12 *for* QVAC *read* QUAC
v. 6, p. 189, l. − 8 *for* CONNOLY *read* CONNOLLY

egrator
hus the

ethod of

Notes
o more
AC on
e hour.
perfect
61, 89,

. L.

A.

B.

C.

D

E.

F.

G

H

I.

J.

K

L

M

N

O

P

Q

R

S

T

U

V

Z

CLASSIFICATION OF TABLES

A. Arithmetical Tables. Mathematical Constants

B. Powers

C. Logarithms

D. Circular Functions

E. Hyperbolic and Exponential Functions

F. Theory of Numbers

G. Higher Algebra

H. Numerical Solution of Equations

I. Finite Differences. Interpolation

J. Summation of Series

K. Statistics

L. Higher Mathematical Functions

M. Integrals

N. Interest and Investment

O. Actuarial Science

P. Engineering

Q Astronomy

R. Geodesy

S. Physics, Geophysics, Crystallography

T. Chemistry

U. Navigation

V. Aerodynamics, Hydrodynamics, Ballistics

Z. Calculating Machines and Mechanical Computation

CONTENTS

JANUARY 1953

Addendum to a Guide to Tables on Punched Cards........ G. BLANCH & E. C. YOWELL 1

The Sieve Problem for All-Purpose Computers........ D. H. L. 6

The Use of Large Intervals in Finite-Difference Equations.... L. FOX 14

Monte Carlo Matrix Inversion and Recurrent Events........ H. P. EDMUNDSON 18

Recent Mathematical Tables........ 21
ALBERT & JOHNSON 1047, BHATE 1048, CALIGO 1059, CAMERON 1049, CHUNG & DELURY 1050, DENGLER, GOLAND, & LUKE 1060, FERRARI 1061, FRÖBERG 1042, GODWIN 1062, HALD & SINKBAEK 1051, KRAITCHIK 1043, LEVENE 1052, MARRIOTT 1053, MORIGUTI 1054, NABEYA 1055, NAIR 1056, NATUCCI 1044, PALAMÀ 1045, QUENOUILLE 1057, SMITH, EDWARDS, & BRINKLEY, 1064, SMOLÎAKOV & HOVANSKIĬ 1046, SUKHATME, THAWANI, PENDHARKAR, & NATU 1058, THIRUVENKATACHAR & RAMAKRISHNA 1063.

Mathematical Tables—Errata........ 31
MCLACHLAN & HUMBERT 218, NBSMTP 219, VAN DER POL 220.

Unpublished Mathematical Tables........ 33
GLODEN 151, 152, 153, GRUENBERGER 154, PORTER 155.

Automatic Computing Machinery........ 34
Technical Developments, The USAF-Fairchild Specialized Digital Computer........ J. J. STONE 35

Discussions, Floating Operations on the EDSAC........ R. A. BROOKER & D. J. WHEELER 37
Bibliography of Coding Procedure........ J. T. 47
Bibliography Z: GARDNER 1000, GOODELL 1001, HARVARD UNIVERSITY 1002, INSTITUTE OF RADIO ENGINEERS 1004, INTERNATIONAL BUSINESS MACHINES CORPORATION 1005–1009, OFFICE OF NAVAL RESEARCH 1003. 48
News........ 57

Other Aids to Computation........ 61
Analogue Calculation of Polynomial and Trigonometric Expansions........ M. G. SCHERBERG & J. F. RIORDAN 61
Bibliography Z: EULER & LUDWIG 1010, GAMBLE & HATTEN 1012, GLUBRECHT 1011, KORN & KORN 1013, TRIBUS 1014, WEINER 1015. 65

Notes........ 67
143. Analytical Approximations........ C. HASTINGS, JR. 67
144. Zeros of the Derivative of Bessel Functions of Fractional Order........ H. E. SALZER 69
145. An Example in the Use of the Differential Analyzer........ J. G. L. MICHEL 71
146. Two New Mersenne Primes........ D. H. L. 72

Corrigenda........ 72

LANCASTER PRESS, INC., LANCASTER, PA.

1
6
14
18
21
31
33
34
35
37
47
48
57
61
61
65
67
67
69
71
72
72
E, PA.